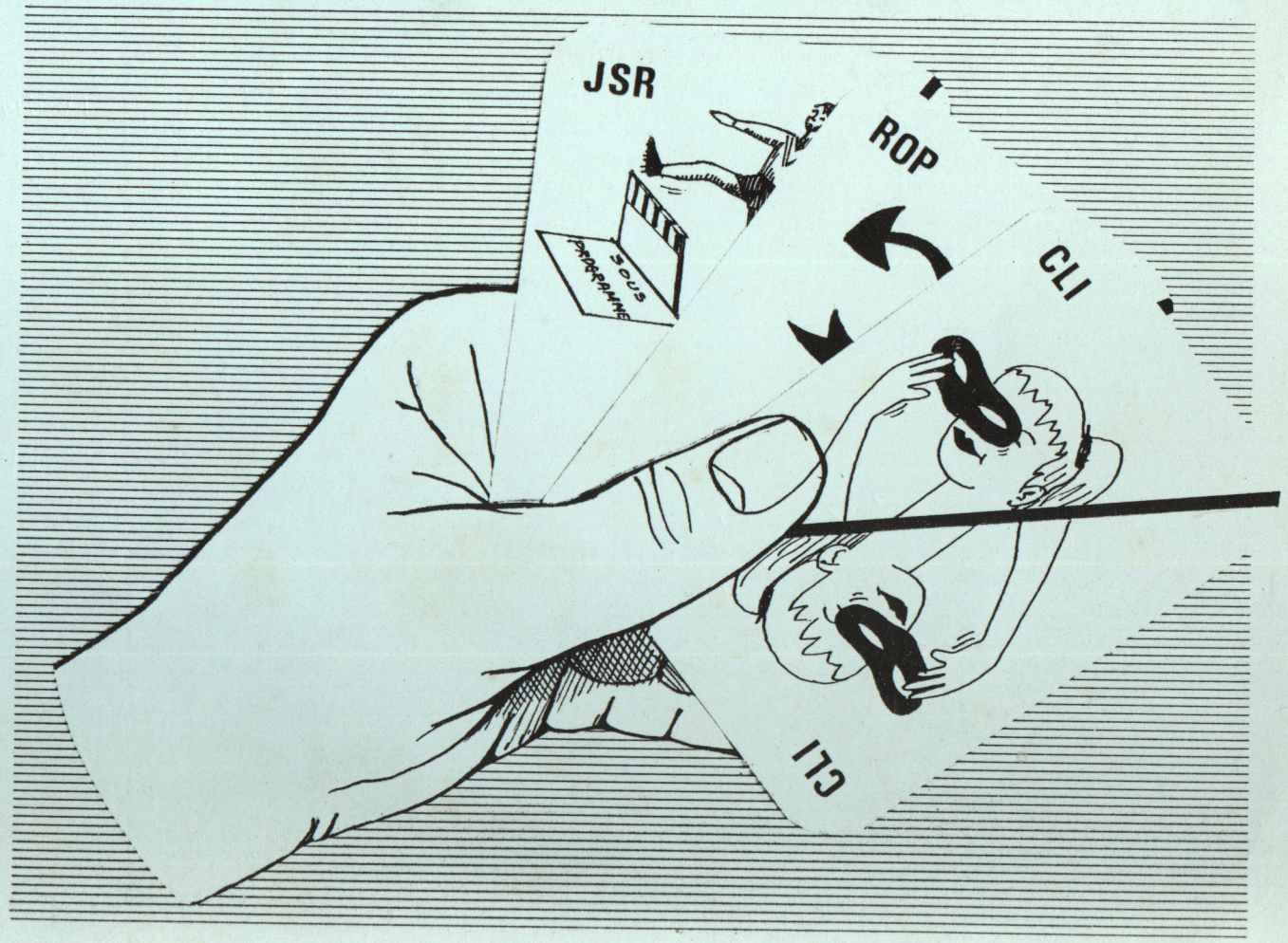
Postes et Télécommunications

Télécommunications

Centre Interrégional d'Enseignement de Lannion

JEU D'INSTRUCTION DU MICROPROCESSEUR MC 6800



DESCRIPTION DU JEU D'INSTRUCTION

MICROPROCESSEUR M C 6800

-=-=-=-=-=-=-=-=-=-=-=-=-=-

ABA	– Addition de l'accumulateur B à l'accumulateur A
ADC	– Addition avec retenue
ADD	– Addition
AND	– « ET » logique
ASL	– Décalage arithmétique gauche
ASR	– Décalage arithmétique droit
BCC	– Branchement si pas de retenue
BCS	– Branchement si retenue
BEQ	– Branchement si égal (à zéro)
BGE	– Branchement si supérieur ou égal à zéro
BGT	– Branchement si plus grand que zéro
BHI	– Branchement si supérieur
BIT	– Test de bits
BLE	– Branchement si inférieur ou égal à zéro
BLS	– Branchement si inférieur ou égal
BLT	– Branchement si inférieur à zéro
BMI	– Branchement si négatif
BNE	– Branchement si non nul
BPL	– Branchement si positif ou nul
BRA	– Branchement inconditionnel
BSR	– Branchement à un sous-programme
BVC	– Branchement si pas de dépassement
BVS	– Branchement si dépassement
CBA	– Comparaison des accumulateurs
CLC	– Mise à zéro du bit de retenue
CLI	– Mise à zéro du masque d'interruption
CLR	– Mise à zéro
CLV	– Comparaison
COM	– Complément à un
CPX	– Comparaison du registre d'index
DAA	– Ajustement décimal sur l'accumulateur A
DEC	– Décrémentation
DES	– Décrémentation du pointeur de pile
DEX	– Décrémentation du registre d'index
EOR	– « OU » exclusif
INC	– Incrémentation
INS	– Incrémentation du pointeur de pile
INX	– Incrémentation du registre d'index
JMP	– Saut inconditionnel
JSR	– Saut à un sous programme
LDA	– Chargement accumulateur
LDS	– Chargement du pointeur de pile
LDX	– Chargement du registre d'index
LSR	– Décalage logique droit
NEG	– Complément à deux (opposé)
NOP	– Pas d'opération
ORA	– « OU » logique
PSH	– Empiler
PUL	– Dépiler
ROL	– Rotation à gauche
ROR	– Rotation à droite
RTI	– Retour d'interruption
RTS	– Retour de sous programme
SBA	– Soustraction entre accumulateurs
SBC	– Soustraction avec retenue
SEC	– Mise à un de la retenue
SEI	– Mise à un du masque d'interruption
SEV	– Mise à un du bit de débordement
STA	– Mise en mémoire d'un accumulateur
STS	– Mise en mémoire du pointeur de pile
STX	– Mise en mémoire du registre d'index
SUB	– Soustraction
SWI	– Interruption programmée
TAB	– Transfert de l'accumulateur A dans l'accumulateur B
TAP	– Transfert de l'accumulateur A dans le registre d'Etat
TBA	– Transfert de l'accumulateur B dans l'accumulateur A
TPA	– Transfert du registre d'Etat dans l'accumulateur A
TST	– Test
TSX	– Transfert du pointeur de pile dans le registre d'index
TXS	– Transfert du registre d'index dans le pointeur de pile
WAI	– Attente d'interruption

DOCUMENT EN PRET

ACCUMULATOR AND MEMORY OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED ≠	DIRECT OP	DIRECT ~	DIRECT ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INTHER OP	INTHER ~	INTHER ≠	BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	COND. CODE REG 5 H	4 I	3 N	2 Z	1 V	0 C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	•	•	↕	↕	R	•
Clear	CLR							6F	7	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	•	•	↕	↕	(1)	(2)
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	(1)	(2)
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	(1)	(2)
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	•	•	↕	↕	↕	(3)
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	•	•	↕	↕	(5)	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	(5)	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	(5)	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A⊕M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B⊕M → B	•	•	↕	↕	R	•
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	•	•	↕	↕	(5)	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	(5)	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	(5)	•
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	•	•	↕	↕	R	•
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3				M	•	•	↕	↕	(6)	↕
	ROLA													49	2	1	A	•	•	↕	↕	(6)	↕
	ROLB													59	2	1	B	•	•	↕	↕	(6)	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	(6)	↕
	RORA													46	2	1	A	•	•	↕	↕	(6)	↕
	RORB													56	2	1	B	•	•	↕	↕	(6)	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	(6)	↕
	ASLA													48	2	1	A	•	•	↕	↕	(6)	↕
	ASLB													58	2	1	B	•	•	↕	↕	(6)	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	(6)	↕
	ASRA													47	2	1	A	•	•	↕	↕	(6)	↕
	ASRB													57	2	1	B	•	•	↕	↕	(6)	↕
Shift Right, Logic	LSR							64	7	2	74	6	3				M	•	•	R	↕	(6)	↕
	LSRA													44	2	1	A	•	•	R	↕	(6)	↕
	LSRB													54	2	1	B	•	•	R	↕	(6)	↕
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M	•	•	↕	↕	R	•
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subract Acmltrs	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R

· Code d'Instruction du 6800

INDEX REGISTER AND STACK POINTER OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED ≠	DIRECT OP	DIRECT ~	DIRECT ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INHER OP	INHER ~	INHER ≠	BOOLEAN/ARITHMETIC OPERATION	5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$(X_H/X_L) - (M/M + 1)$	•	•	(7)	↕	(8)	•
Decrement Index Reg	DEX													09	4	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \to X_H$, $(M + 1) \to X_L$	•	•	(9)	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \to SP_H$, $(M + 1) \to SP_L$	•	•	(9)	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \to M$, $X_L \to (M + 1)$	•	•	(9)	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \to M$, $SP_L \to (M + 1)$	•	•	(9)	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	SP + 1 → X	•	•	•	•	•	•

JUMP AND BRANCH OPERATIONS	MNEMONIC	RELATIVE OP	RELATIVE ~	RELATIVE ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INHER OP	INHER ~	INHER ≠	BRANCH TEST	5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If > Zero	BGE	2C	4	2										N⊕V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										Z + (N⊕V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If < Zero	BLE	2F	4	2										Z + (N⊕V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										N⊕V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2											•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				} See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3					•	•	•	•	•	•
No Operation	NOP										01	2	1	Ad Prog Cntr. Only	•	•	•	•	•	•
Return From interrupt	RTI										3B	10	1		———	———	(10)	———	———	———
Return From Subroutine	RTS										39	5	1	} See special Operation	•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI										3E	9	1		•	(11)	•	•	•	•

CONDITIONS CODE REGISTER OPERATIONS	MNEMONIC	INHER OP	INHER ~	INHER ≠	BOOLEAN OPERATION	5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	A → CCR	———	———	(12)	———	———	———
CCR → Acmltr A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

LEGEND

OP	Operation Code (Hexadecimal)	00	Byte = Zero
~	Number of MPU Cycles	H	Half carry from bit 3
≠	Number of Program Bytes	I	Interrupt mask
+	Arithmetic Plus	N	Negative (sign bit)
−	Arithmetic Minus	Z	Zero (byte)
·	Boolean AND	V	Overflow, 2's complement
M_{SP}	Contents of memory location pointed to be Stack Pointer	C	Carry from bit 7
		R	Reset Always
+	Boolean Inclusive OR	S	Set Always
⊕	Boolean Exclusive OR	↕	Test and set if true cleared otherwise
$\overline{M}$	Complement of M	•	Not Affected
→	Transfer Into	CCR	Condition Code Register
0	Bit = Zero	LS	Least Significant
		MS	Most Significant

CONDITION CODE REGISTER NOTES

(Bit set if test is true and cleared otherwise)

(1) (Bit V) Test: Result = 10000000 ?

(2) (Bit C) Test: Result = 00000000 ?

(3) (Bit C) Test: Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)

(4) (Bit V) Test: Operand = 10000000 prior to execution ?

(5) (Bit V) Test: Operand = 01111111 prior to execution ?

(6) (BIT V) Test: Set equal to result of N⊕C after shift has occurred

(7) (Bit N) Test: Sign bit of most significant (MS) byte of result = 1 ?

(8) (Bit V) Test: 2's complement overflow from subtraction of LS bytes ?

(9) (Bit N) Test: Result less than zero ? (Bit 15 = 1)

(10) (All) Load Condition Code Register from Stack. (See Special Operations)

(11) (Bit I) Set when interrupt occurs. If previously set, a Non Maskable interrupt is required to exit the wait state.

(12) (ALL) Set according to the contents of Accumulator A

Page inversée

ABA — Addition de B à A

Fonction :

A ← (A) + (B)

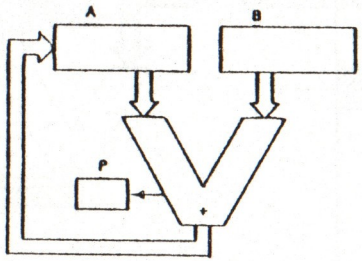
Format :

00011011

Description :

Ajoute les contenus des deux accumulateurs A et B. Le résultat est déposé dans A.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent

HEX = 1B, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT	●		●	●	●	●

ADD

Addition avec retenue

Fonction :

R ← (R) + DONNEE + C

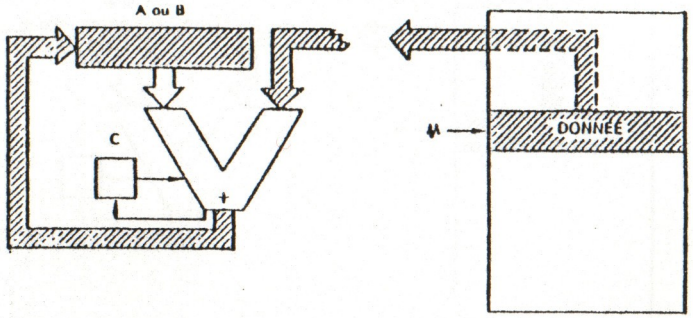
Format :

1abb1011	ADR/DON	ADR

Description :

Ajoute à l'accumulateur spécifié le contenu de l'adresse-mémoire ou du littéral. Le résultat est laissé dans l'accumulateur.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				8B	9B	AB	BB		CB	DB	EB	FB
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	6	4		2	3	5	4
abb				000	001	010	011		100	101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT	●		●	●	●	●

Codes opération :

A	IMMEDIAT	10001011	DON	abb = 000 HEX = 8B CYCLES = 2
A	PAGE 0	10011011	ADR	abb = 001 HEX = 9B CYCLES = 3
A	INDEXE	10101011	DEPL	abb = 010 HEX = AB CYCLES = 6
A	ABSOLU	10111011	ADRESSE 16 BITS	abb = 011 HEX = BB CYCLES = 4
B	IMMEDIAT	11001011	DON	abb = 100 HEX = CB CYCLES = 2
B	PAGE 0	11011011	ADR	abb = 101 HEX = DB CYCLES = 3
B	INDEXE	11101011	DEPL	abb = 110 HEX = EB CYCLES = 6
B	ABSOLU	11111011	ADRESSE 16 BITS	abb = 111 HEX = FB CYCLES = 4

ADC

Addition avec retenue

Fonction :

R ← (R) + DONNEE + C

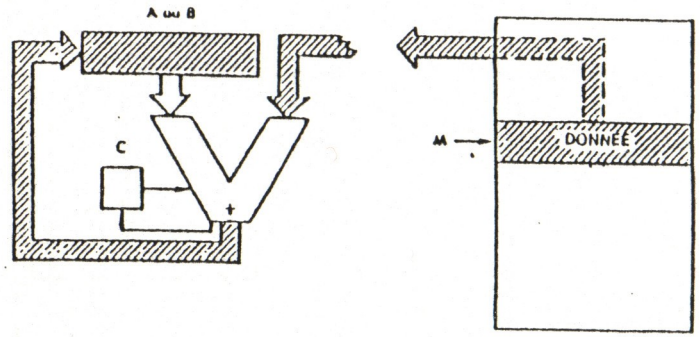
Format :

bbb1001	ADH/DON	ADR

Description :

Ajoute à l'accumulateur spécifié le contenu de l'adresse-mémoire ou du littéral plus le bit de retenue. Le résultat est laissé dans l'accumulateur.

Circuit des informations :



Modes d'adressage :

	IMM	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				89	99	A9	B9		C9	D9	E9	F9
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	6	4		2	3	6	4
abb				000	001	010	011		100	101	110	111

INDICATEURS D'ETAT	H	I	N	Z	V	C
	●		●	●	●	●

Codes opération :

A	IMMEDIAT	10001001	DON	
	abb = 000	HEX = 89	CYCLES = 2	
A	PAGE 0	10011001	ADR	
	abb = 001	HEX = 99	CYCLES = 3	
A	INDEXE	10101001	DEPL	
	abb = 010	HEX = A9	CYCLES = 6	
A	ABSOLU	10111001	ADRESSE 16 BITS	
	abb = 011	HEX = B9	CYCLES = 4	
B	IMMEDIAT	11001001	DON	
	abb = 100	HEX = C9	CYCLES = 2	
B	PAGE 0	11011001	ADR	
	abb = 101	HEX = D9	CYCLES = 3	
B	INDEXE	11101001	DEPL	
	abb = 110	HEX = E9	CYCLES = 6	
B	ABSOLU	11111001	ADRESSE 16 BITS	
	abb = 111	HEX = F9	CYCLES = 4	

AND

ET logique

Fonction :

R ← (R) ∧ DONNEE

Format :

1 abb 0100	ADR/DONNEE	ADR

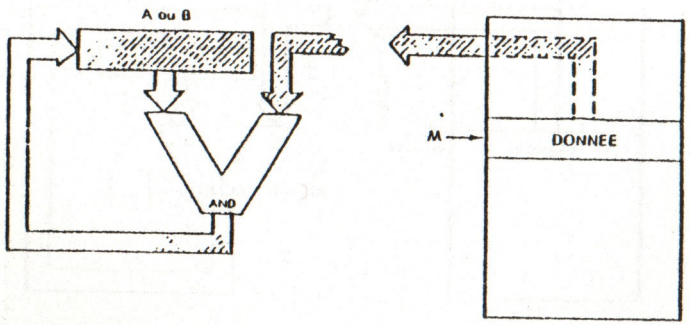
Description :

Effectue le ET logique entre un accumulateur et la donnée indiquée. Le résultat est déposé dans l'accumulateur.

La table de vérité est

A\M	0	1
0	0	0
1	0	1

Circuit des informations



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				84	94	A4	B4		C4	D4	E4	F4
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	6	4		2	3	6	4
abb				000	001	010	011		100	101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	○	

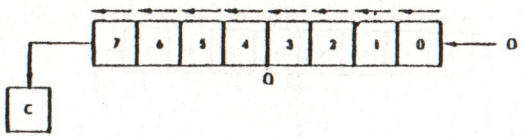
Codes opération :

A	IMMEDIAT	10000100	DON	abb = 000 HEX = 84 CYCLES = 2
A	PAGE 0	10010100	ADR	abb = 001 HEX = 94 CYCLES = 3
A	INDEXE	10100100	DEPL	abb = 010 HEX = A4 CYCLES = 6
A	ABSOLU	10110100	ADRESSE 16 BITS	abb = 011 HEX = B4 CYCLES = 4
B	IMMEDIAT	11000100	DON	abb = 100 HEX = C4 CYCLES = 2
B	PAGE 0	11010100	ADR	abb = 101 HEX = D4 CYCLES = 3
B	INDEXE	11100100	DEPL	abb = 110 HEX = E4 CYCLES = 6
B	ABSOLU	11110100	ADRESSE 16 BITS	abb = 111 HEX = F4 CYCLES = 4

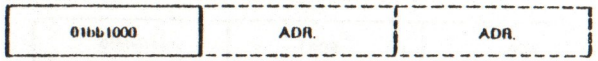
ASL

Décalage arithmétique gauche

Fonction :



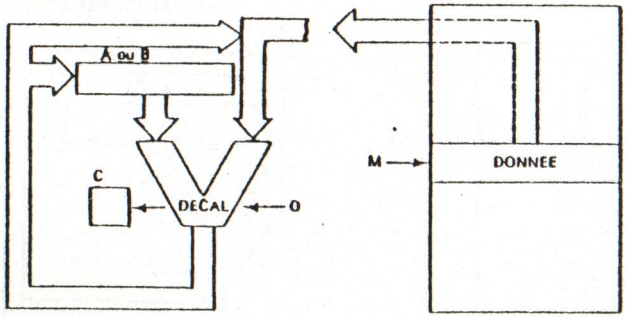
Format :



Description :

Décale le contenu de l'accumulateur ou de l'emplacement mémoire d'un bit vers la gauche. Il entre un 0 sur la droite. Le bit 7 tombe dans la retenue. Le résultat est déposé dans la source : accumulateur ou mémoire.

Circuit des informations



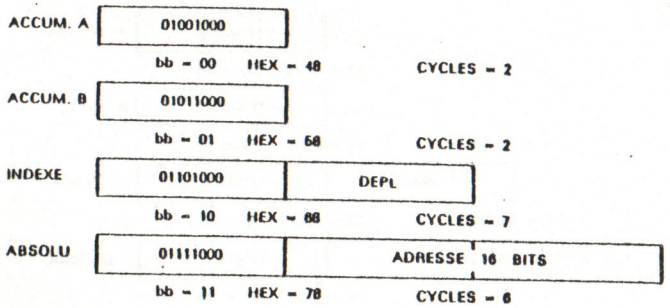
Modes d'adressage

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			48			68	78	58				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	*	●

* V est mis à N ⊕ C après décalage.

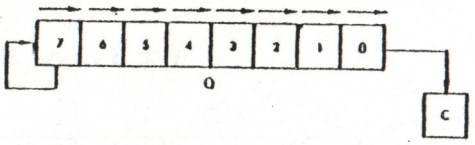
Codes opération :



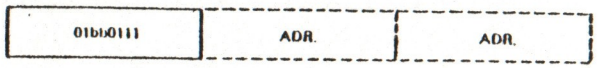
ASR

Décalage arithmétique droite

Fonction :



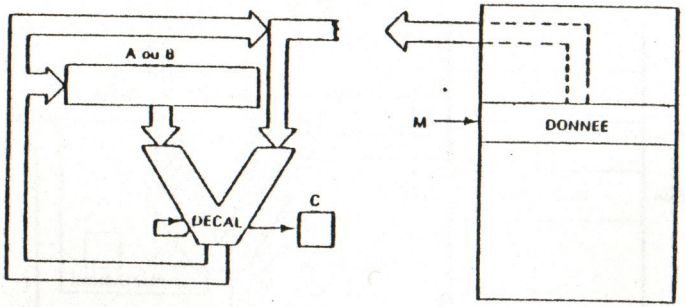
Format :



Description :

Décale le contenu de l'accumulateur ou de l'emplacement mémoire d'un bit vers la droite en répétant le bit de signe. Le bit 0 tombe dans la retenue. Le résultat est déposé dans la source : accumulateur ou mémoire.

Circuit des informations



Modes d'adressage

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			47			67	77	67				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	*	●

* V est mis à N ⊕ C après décalage

Codes opération :

ACCUM. A	01000111		
	bb = 00	HEX = 47	CYCLES = 2
ACCUM. B	01010111		
	bb = 01	HEX = 57	CYCLES = 2
INDEXE	01100111	DEPL	
	bb = 10	HEX = 67	CYCLES = 7
ABSOLU	01110111	ADRESSE 16 BITS	
	bb = 11	HEX = 77	CYCLES = 6

BCC — Branchement si pas de retenue

Fonction :

Va à l'adresse indiquée si C = 0.

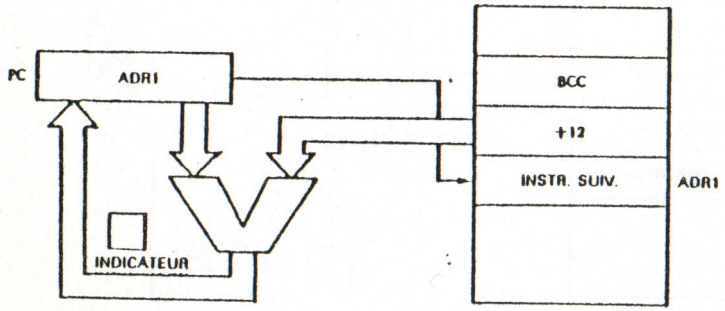
Format :

00100100	DEPLACEMENT

Description :

On teste l'indicateur de retenue. Si C = 0 on se branche à adresse courante plus déplacement signé (entre — 128 et + 127). Si C = 1, on ne fait rien. Le déplacement est ajouté à l'adresse de l'instruction suivant le BCC. Il en résulte un déplacement effectif entre — 126 et + 129.

Circuit des informations :



Mode d'adressage :

Uniquement relatif.

HEX = 24, octets = 2, cycles = 4

INDICATEURS D'ETAT	H	I	N	Z	V	C

(AUCUN CHANGEMENT)

BCS

Branchement si retenue

Fonction :

Va à l'adresse indiquée si C = 1.

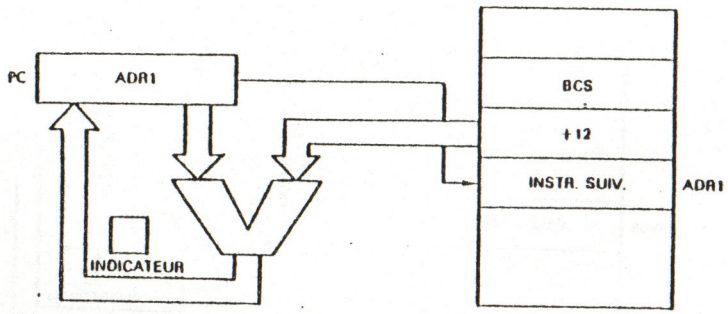
Format :

00100101	DEPLACEMENT

Description :

On teste l'indicateur de retenue. Si C = 1, on se branche à adresse courante plus le déplacement signé (de — 128 à + 127). Si C = 0, on ne fait rien. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BCS. Il en résulte un déplacement effectif de — 126 à + 129.

Circuit des informations



Mode d'adressage :

Uniquement relatif.

HEX = 25, octets = 2, cycles = 4

INDICATEURS D'ETAT	H	I	N	Z	V	C

(AUCUN CHANGEMENT)

BEQ

Branchement si égalité

BEQ

Fonction :

Va à l'adresse indiquée si Z = 1 (c'est-à-dire résultat = 0).

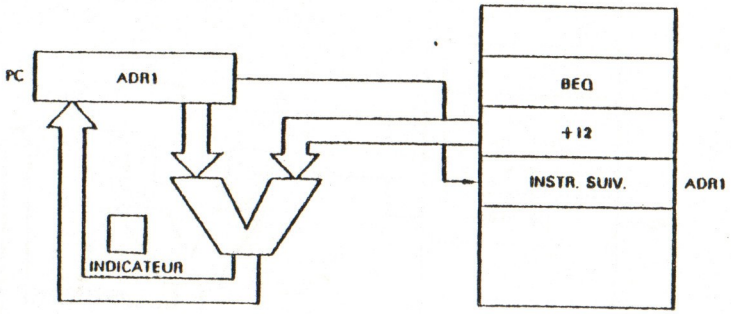
Format :

00100111	DEPLACEMENT

Description :

On teste l'indicateur Z. Si Z = 1, on se branche à adresse courante + déplacement signé (de — 128 à + 127). Si Z = 0, on ne fait rien c'est-à-dire qu'on passe à l'instruction suivante. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BEQ. Il en résulte un déplacement effectif de — 126 à + 129.

Circuit des informations



Mode d'adressage :

Uniquement relatif

HEX = 27, octets = 2, cycles = 4

INDICATEURS D'ETAT	H	I	N	Z	V	C

(AUCUN CHANGEMENT)

BGE

Branchement si supérieur ou égal en complément à 2

Fonction :

Va à l'adresse indiquée si $N \oplus V = 0$.

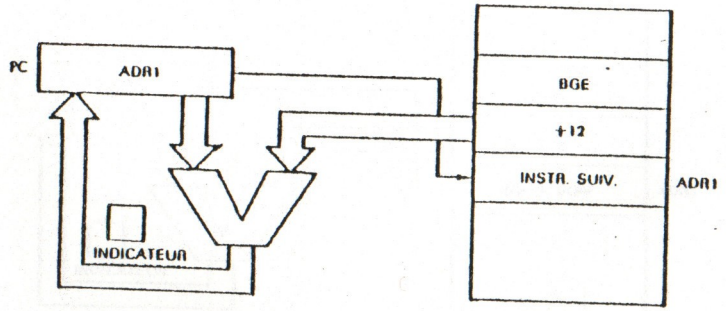
Format :

00101100	DEPLACEMENT

Description :

On teste si la dernière comparaison effectuée a donné $R \geqslant M$ en complément à deux (condition $N \oplus V = 0$). Si oui, on se branche à adresse courante plus déplacement signé (entre − 128 et + 127). Sinon, on passe à l'instruction suivante. Le déplacement est ajouté à l'adresse de l'instruction suivant le BGE. Il en résulte un déplacement effectif entre − 126 et + 129.

Circuit des informations :



Mode d'adressage :

Uniquement relatif.

HEX = 26, octets = 2, cycles = 4

INDICATEURS D'ETAT	H	I	N	Z	V	C

(AUCUN CHANGEMENT)

BGT

Branchement si supérieur en complément à deux

Fonction :

Va à l'adresse indiquée si $Z \vee (N \oplus V) = 0$

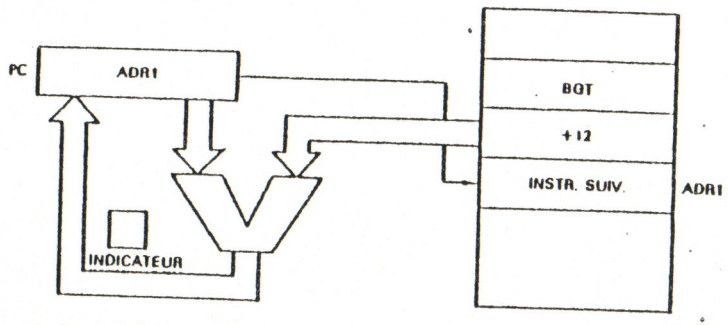
Format :

00101110	DEPLACEMENT

Description :

On teste si la dernière comparaison effectuée a donné $R > M$ en complément à deux (condition $Z \vee (N \oplus V) = 0$). Si oui, on se branche à adresse courante plus déplacement signé (entre − 128 et + 127). Sinon, on passe à l'instruction suivante. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BGT. Il en résulte un déplacement effectif entre − 126 et + 129.

Circuit des informations



Mode d'adressage :

Uniquement relatif.

HEX = 2E, octets = 2, cycles = 4

INDICATEURS D'ETAT	H	I	N	Z	V	C

(AUCUN CHANGEMENT)

BHI

Branchement si supérieur (nombres sans signe)

Fonction :

Va à l'adresse indiquée si C v Z = 0.

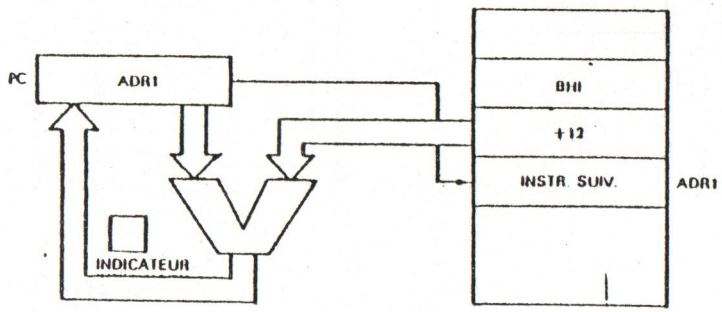
Format :

00100010	DEPLACEMENT

Description :

On teste si la dernière comparaison effectuée a donné R > M (nombres non signés) (condition C v Z= 0). Si oui, on se branche à adresse courante plus déplacement signé (entre − 128 et + 127). Sinon, on passe à l'instruction suivante. Le déplacement est ajouté à l'adresse de l'instruction suivant le BHI. Il en résulte un déplacement effectif entre − 126 et + 129.

Circuit des informations :



Mode d'adressage :

Uniquement relatif.

HEX = 22, octets = 2, cycles = 4

INDICATEURS D'ETAT	H	I	N	Z	V	C

(AUCUN CHANGEMENT)

BIT

Test de bits en mémoire

Fonction :

$Z \leftarrow \overline{(R) \wedge (M)}$, $N \leftarrow R_7 \wedge M_7$, $V \leftarrow 0$

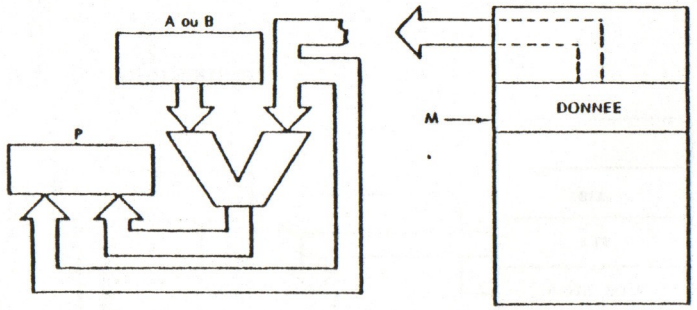
Format :

1abb0101	ADR	ADR

Description :

On effectue le ET entre un accumulateur et M mais le résultat n'est pas déposé dans l'accumulateur. Le résultat est indiqué par Z : Z = 1 s'il y a accord, 0 sinon. Le contenu de l'accumulateur reste inchangé.

Circuit des informations



Modes d'adressage

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				85	95	A5	B5		C5	D5	E5	F5
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	5	4		2	3	5	4
abb				000	001	010	011		100	101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	○	

Codes opération :

A	IMMEDIAT	10000101	DON	
		abb = 000	HEX = 85	CYCLES = 2
A	PAGE 0	10010101	ADR	
		abb = 001	HEX = 95	CYCLES = 3
A	INDEXE	10100101	DEPL	
		abb = 010	HEX = A5	CYCLES = 5
A	ABSOLU	10110101	ADRESSE 16 BITS	
		abb = 011	HEX = B5	CYCLES = 4
B	IMMEDIAT	11000101	DON	
		abb = 100	HEX = C5	CYCLES = 2
B	PAGE 0	11010101	ADR	
		abb = 101	HEX = D5	CYCLES = 3
B	INDEXE	11100101	DEPL	
		abb = 110	HEX = E5	CYCLES = 5
B	ABSOLU	11110101	ADRESSE 16 BITS	
		abb = 111	HEX = F5	CYCLES = 4

BLE

Branchement si inférieur ou égal (complément à deux)

Fonction :

Va à l'adresse indiquée si $Z \vee (N \oplus V) = 1$.

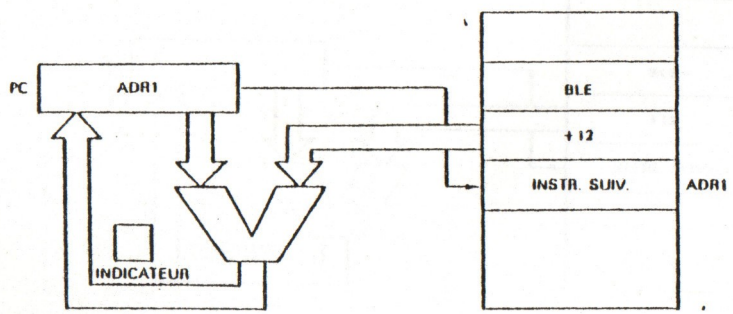
Format :

00101111	DEPLACEMENT

Description :

On teste si la dernière comparaison effectuée a donné $R \leqslant M$ en complément à deux (condition $Z \vee (N \oplus V) = 1$). Si oui, on se branche à adresse courante plus déplacement signé (entre − 128 et + 127). Sinon, on passe à l'instruction suivante. Le déplacement est ajouté à l'adresse de l'instruction suivant le BLE. Il en résulte un déplacement effectif entre − 126 et + 129.

Circuit des informations



Mode d'adressage :

Uniquement relatif.

HEX = 2F, octets = 2, cycles = 4

	H	I	N	Z	V	C
INDICATEURS D'ETAT						

(AUCUN CHANGEMENT)

BLS

Branchement si inférieur ou égal (nombres sans signe)

Fonction :

Va à l'adresse indiquée si C ∨ Z = 1.

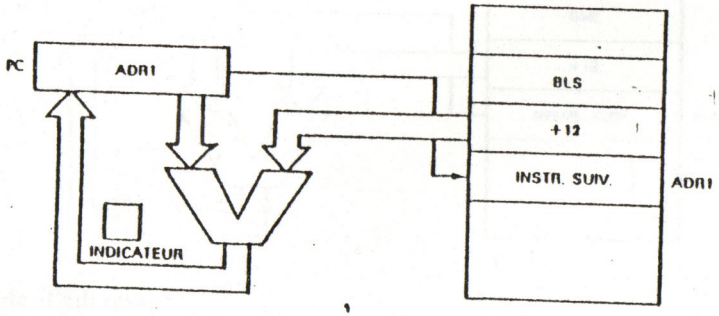
Format :

00100011	DEPLACEMENT

Description :

On teste si la dernière comparaison effectuée a donné R ⩽ M (nombres sans signe) (condition C ∨ Z = 1). Si oui, on se branche à adresse courante plus déplacement signé (entre − 128 et + 127). Sinon, on passe à l'instruction suivante. Le déplacement est ajouté à l'adresse de l'instruction suivant le BLS. Il en résulte un déplacement effectif entre − 126 et + 129.

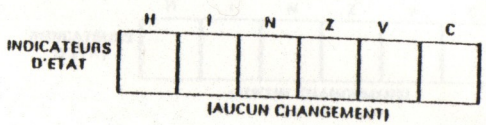
Circuit des informations :



Mode d'adressage :

Uniquement relatif.

HEX = 23, octets = 2, cycles = 4



BLT

Branchement si inférieur (complément à deux)

Fonction :

Va à l'adresse indiquée si N ⊕ V = 1.

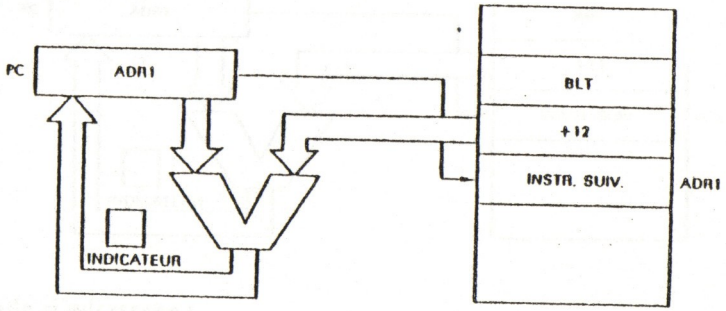
Format :

00101101	DEPLACEMENT

Description :

On teste si la dernière comparaison effectuée a donné R < M en complément à deux (condition N ⊕ V = 1). Si oui, on se branche à adresse courante plus déplacement signé (entre − 128 et + 127). Sinon, on passe à l'instruction suivante. Le déplacement est ajouté à l'adresse de l'instruction suivant le BLT. Il en résulte un déplacement effectif entre − 126 et + 129.

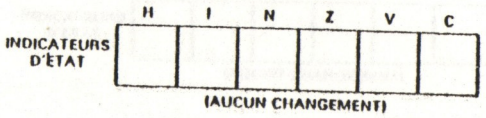
Circuit des informations



Mode d'adressage :

Uniquement relatif.

HEX = 2D, octets = 2, cycles = 4



BMI

Branchement si moins

Fonction :

Va à l'adresse indiquée si N = 1 (résultat < 0).

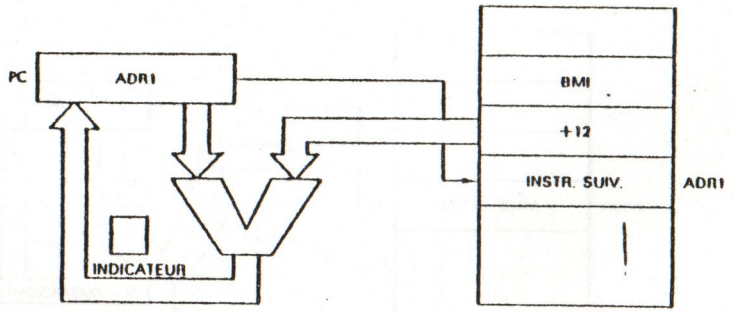
Format :

00101011	DEPLACEMENT

Description :

On teste l'indicateur N (signe). Si N = 1, on se branche à l'adresse courante plus le déplacement signé (de — 128 à + 127). Si N = 0, on continue en séquence. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BMI. Il en résulte un déplacement effectif de — 126 à + 129.

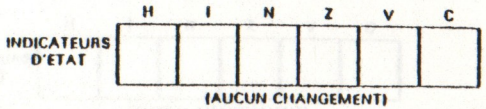
Circuit des informations :



Mode d'adressage :

Uniquement relatif.

HEX = 2B, octets = 2, cycles = 4



BNE

Branchement si non égal à zéro

Fonction :

Va à l'adresse indiquée si Z = 0 (résultat ≠0).

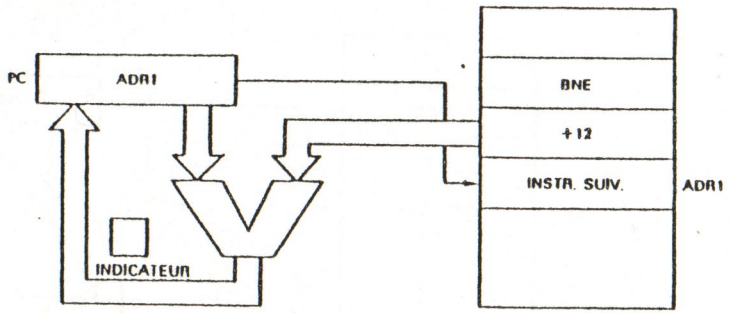
Format :

00100110	DEPLACEMENT

Description :

Teste le bit Z. Si le résultat n'est pas égal à 0 (Z = 0), on se branche à adresse courante + déplacement signé (de — 128 à + 127). Si Z = 1, on continue en séquence. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BNE. Il en résulte un déplacement effectif de — 126 à + 129.

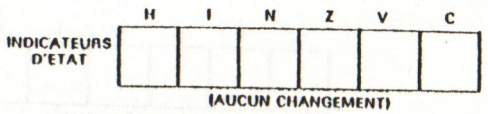
Circuit des informations :



Mode d'adressage :

Uniquement relatif

HEX = 26, octets = 2, cycles = 4



BPL

Branchement si plus

Fonction :

Va à l'adresse indiquée si N = 0 (Résultat > 0)

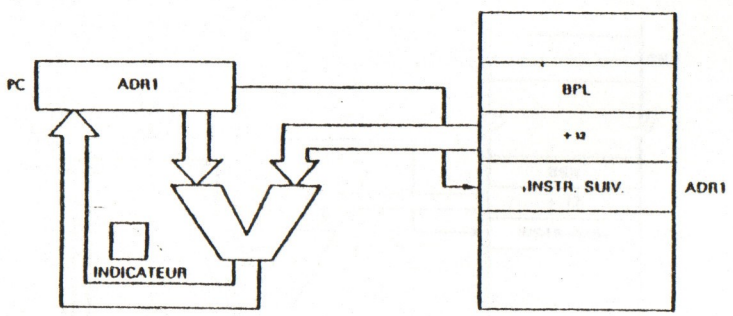
Format :

00101010	DEPLACEMENT

Description :

On teste l'indicateur N (signe). Si N = 0 (résultat positif), on se branche à adresse courante plus déplacement signé (de — 128 à + 127). Si N = 1, on continue en séquence. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BPL. Il en résulte un déplacement effectif de — 126 à + 129.

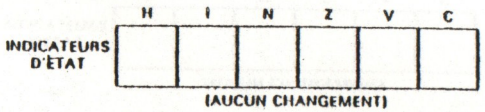
Circuit des informations :



Mode d'adressage :

Uniquement relatif

HEX = 2A, octets = 2, cycles = 4



BRA

Branchement inconditionnel (adressage relatif)

Fonction :

Va à l'adresse indiquée dans tous les cas :
PC ← PC + DEPL

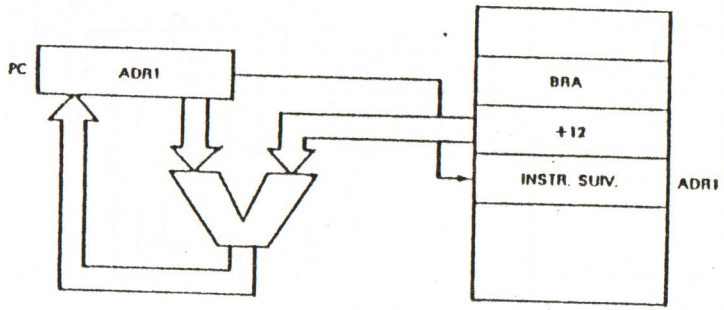
Format :

00100000	DEPLACEMENT

Description :

On se branche inconditionnellement à adresse courante plus déplacement signé (entre − 128 et + 127). Le déplacement est ajouté à l'adresse de l'instruction suivant le BRA. Il en résulte un déplacement effectif entre − 126 et + 129.

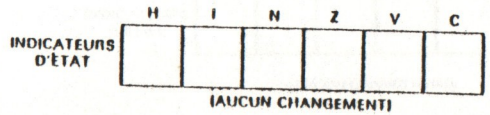
Circuit des informations :



Mode d'adressage :

Uniquement relatif.

HEX = 20, octets = 2, cycles = 4



BSR — Branchement à un sous-programme (adressage relatif)

Fonction :

EMPILE PC puis :
PC ← PC + DEPL

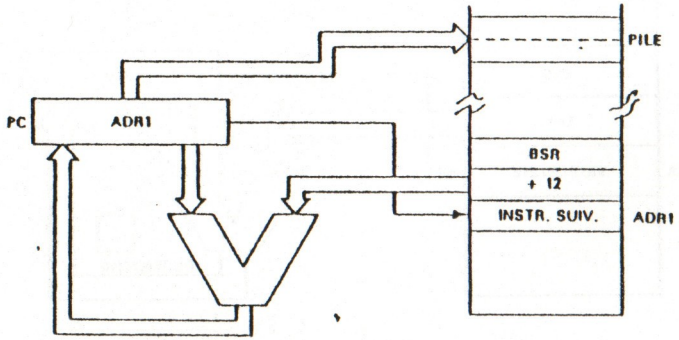
Format :

10001101	DEPLACEMENT

Description :

On empile PC, puis on se branche à adresse courante plus le déplacement signé (de − 128 à + 127). Le déplacement est ajouté à l'adresse de l'instruction qui suit le BSR. Il en résulte un déplacement effectif de − 126 à + 129. C'est avantageux pour appeler un sous-programme situé assez près : on gagne un octet.

Circuit des informations



Mode d'adressage :

Uniquement relatif.

HEX = 8D, octets = 2, cycles = 8

	H	I	N	Z	V	C
INDICATEURS D'ETAT						

(AUCUN CHANGEMENT)

BVC — Branchement si pas de débordement

Fonction :

Va à l'adresse indiquée si V = 0

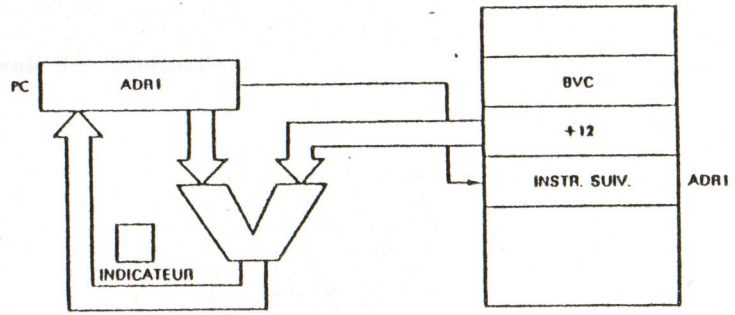
Format :

00101000	DEPLACEMENT

Description :

On teste l'indicateur de débordement (V). Si V = 0 (pas de débordement), on se branche à adresse courante plus déplacement signé (de — 128 à + 127). Si V = 1 on continue en séquence. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BVC. Il en résulte un déplacement effectif de — 126 à + 129.

Circuit des informations :



Mode d'adressage :

Uniquement relatif

HEX = 28, octets = 2, cycles = 4

	H	I	N	Z	V	C
INDICATEURS D'ETAT						

(AUCUN CHANGEMENT)

BVS — Branchement si débordement

Fonction :

Va à l'adresse indiquée si V = 1

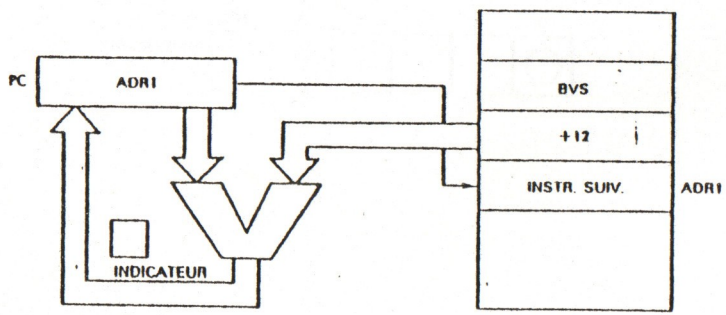
Format :

00101001	DEPLACEMENT

Description :

On teste l'indicateur de débordement (V). S'il y a eu un débordement (V = 1), on se branche à adresse courante plus déplacement signé (de — 128 à + 127). Si V = 0, on continue en séquence. Le déplacement est ajouté à l'adresse de l'instruction qui suit le BVS. Il en résulte un déplacement effectif de — 126 à + 129.

Circuit des informations :



Mode d'adressage :

Uniquement relatif

HEX = 29, octets = 2, cycles = 4

	H	I	N	Z	V	C
INDICATEURS D'ETAT						

(AUCUN CHANGEMENT)

CBA — Comparaison des deux accumulateurs

Fonction :

(A) – (B)

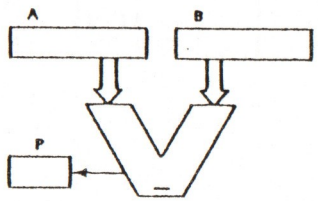
Format :

00010001

Description :

L'accumulateur B est soustrait de A. Le résultat n'est pas rangé de sorte que A et B restent inchangés, mais les indicateurs sont positionnés : N est mis à 1 si le résultat est négatif ; Z est mis à 1 si le résultat est nul (donc A = B) ; V est mis à 1 si la soustraction produit un débordement ; C est mis à 1 si, considérant les nombres comme non signés, B > A. CBA est généralement suivi d'un branchement BHI ou BLS (nombres signés), BGE, BGT, BLT (nombres en complément à 2).

Circuit des informations :



Mode d'adressage :

Uniquement inhérent.

HEX = 11, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	●	●

CLC — Mise à zéro de la retenue

Fonction :

C ← 0

Format :

00001100

Description :

On met à 0 le bit de retenue. C'est souvent nécessaire avant un ADC.

Mode d'adressage :

Uniquement inhérent
HEX = 0C, octet = 1, cycles = 2

INDICATEURS D'ETAT	H	I	N	Z	V	C
						0

CLI — Mise à zéro du masque des interruptions

Fonction :

I ← 0

Format :

00001110

Description :

On met à 0 le masque des interruptions. Ceci autorise les interruptions. Une routine de traitement d'interruption doit toujours mettre I à 0 sinon d'autres interruptions risquent d'être perdues.

Mode d'adressage :

Uniquement inhérent
HEX = 0E, octet = 1, cycles = 2

INDICATEURS D'ETAT	H	I	N	Z	V	C
		0				

CLR — Mise à zéro

Fonction :

Q ← 0

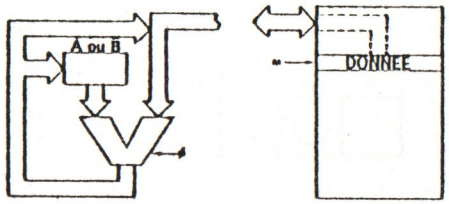
Format :

01bb1111	ADR.	ADR.

Description :

Met à zéro le contenu de l'accumulateur ou de l'emplacement mémoire spécifié.

Circuit des informations



Modes d'adressage

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			4F			6F	7F	5F				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

INDICATEURS D'ETAT	H	I	N	Z	V	C
			O	1	O	O

Codes opération :

ACCUM. A: 01001111
bb = 00 HEX = 4F CYCLES = 2

ACCUM. B: 01011111
bb = 01 HEX = 5F CYCLES = 2

INDEXE: 01101111 | DEPL
bb = 10 HEX = 6F CYCLES = 7

ABSOLU: 01111111 | ADRESSE 16 BITS
bb = 11 HEX = 7F CYCLES = 6

CLV Mise à zéro de l'indicateur de débordement

Fonction :

V ← 0

Format :

00001010

Description :

L'indicateur de débordement est mis à zéro.

Mode d'adressage :

Uniquement inhérent
HEX = 0A, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT					0	

CMP

Comparer à un accumulateur

Fonction :

(R) — DONNEE ⇒ NZCV

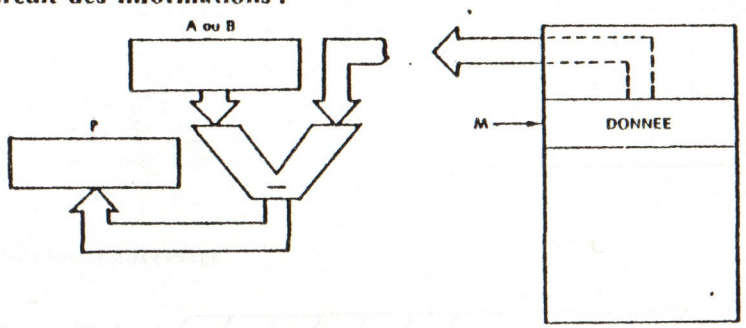
Format :

1abb0001	ADR/DONNEE	ADR

Description :

La donnée indiquée est soustraite de l'accumulateur spécifié. Le résultat n'est pas rangé dans l'accumulateur mais les indicateurs N, Z, V et C sont positionnés : N est mis à 1 si le résultat est négatif ; Z est mis à 1 si le résultat est nul (donc si les deux termes de la comparaison sont égaux) ; V est mis à 1 si la soustraction produit un débordement ; C est mis à 1 si, lorsque l'on considère les nombres comme non signés, (M) est supérieur à (R). CMP est généralement suivi d'un branchement BHI ou BLS si l'on considère les nombres comme non signés, BGE, BGT, BLE, BLT si on les considère comme en complément à deux.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				81	91	A1	B1		C1	D1	E1	F1
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	6	4		2	3	6	4
abb				000	001	010	011		100	101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	●	●

Codes opération :

A	IMMEDIAT	10000001	DON	abb = 000 HEX = 81 CYCLES = 2
A	PAGE 0	10010001	ADR	abb = 001 HEX = 91 CYCLES = 3
A	INDEXE	10100001	DEPL	abb = 010 HEX = A1 CYCLES = 6
A	ABSOLU	10110001	ADRESSE 16 BITS	abb = 011 HEX = B1 CYCLES = 4
B	IMMEDIAT	11000001	DON	abb = 100 HEX = C1 CYCLES = 2
B	PAGE 0	11010001	ADR	abb = 101 HEX = D1 CYCLES = 3
B	INDEXE	11100001	DEPL	abb = 110 HEX = E1 CYCLES = 6
B	ABSOLU	11110001	ADRESSE 16 BITS	abb = 111 HEX = F1 CYCLES = 4

COM

Complémentation (à 1)

Fonction :

$Q \leftarrow \bar{Q}$

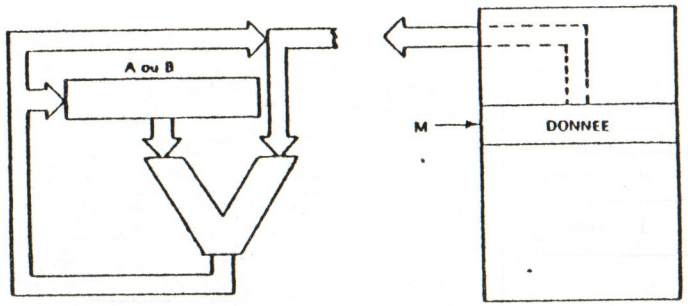
Format :

01bb0011	ADR.	ADR.

Description :

Complémente (à 1) le contenu de l'accumulateur ou de l'emplacement mémoire indiqué : chaque bit est inversé.

Circuit des informations



Modes d'adressage

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			43			63	73	53				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			11	10	01				

INDICATEURS D'ETAT	H	I	N	Z	V	C
			●	●	○	1

Codes opération :

ACCUM. A — 01000011
bb = 00 HEX = 43 CYCLES = 2

ACCUM. B — 01010011
bb = 01 HEX = 53 CYCLES = 2

INDEXE — 01100011 | DEPL
bb = 10 HEX = 63 CYCLES = 7

ABSOLU — 01110011 | ADRESSE 16 BITS
bb = 11 HEX = 73 CYCLES = 6

CPX

Comparer à IX

Fonction :

(IXH, IXL) – (M, M') ⇒ N Z V

Format :

10bb1100	ADR/DONNEE	ADR

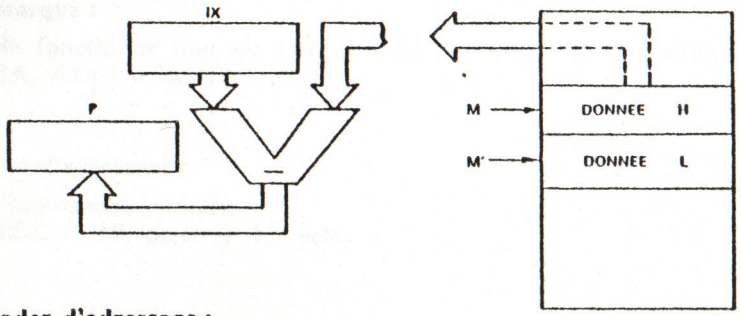
Description :

La donnée indiquée (partie haute en M, partie basse à l'adresse suivante) est soustraite de IX (16 bits) mais le résultat n'est pas rangé dans IX. Seuls les indicateurs sont positionnés.

Z est mis à 1 s'il y a égalité sur 16 bits et il peut être utilisé dans un branchement conditionnel subséquent.

N et V sont modifiés mais ne peuvent être utilisés dans un branchement conditionnel car ils ne dépendent que de la soustraction des octets de poids fort.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				BC	9C	AC	BC					
OCTETS				3	2	2	3					
CYCLES				3	4	6	6					
bb				00	01	10	11					

INDICATEURS D'ETAT	H	I	N	Z	V	C
			●	●	●	

Codes opération :

IMMEDIAT	10001100	DONNEE	16 BITS
	bb = 00	HEX = 8C	CYCLES = 3
PAGE 0	10011100	ADR	
	bb = 01	HEX = 9C	CYCLES = 4
INDEXE	10101100	DEPL	
	bb = 10	HEX = AC	CYCLES = 6
ABSOLU	10111100	ADRESSE	16 BITS
	bb = 11	HEX = BC	CYCLES = 6

DAA

Ajustement décimal de A

Fonction :

$$A \leftarrow (A) + \begin{Bmatrix} 00 \\ 06 \\ 60 \\ 66 \end{Bmatrix}$$

Format :

00011001

Description :

Ajuste l'accumulateur A pour obtenir un résultat de l'addition précédente conforme à l'arithmétique DCB. N est mis à 1 si le bit 7 du résultat est à 1 (ne signifie rien ici) ; Z est mis à 1 si le résultat est nul ; V est indéfini ; C est mis à 1 si le résultat est > 99 (correct).

Remarque :

Ne fonctionne que sur l'accumulateur A et après une addition (ABA, ADC ou ADD).

Mode d'adressage :

Uniquement inhérent
HEX = 19, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	●	●

DEC

Décrémenter

Fonction :

Q ← (Q) − 1

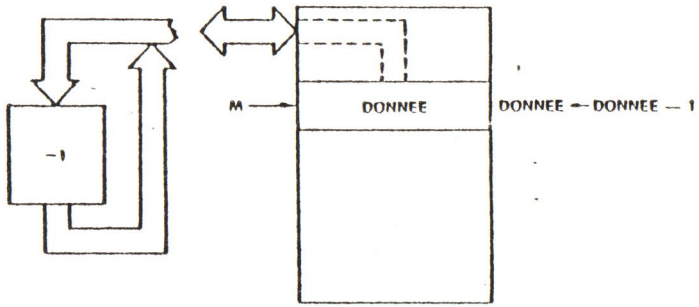
Format :

01bb1010	ADR	ADR

Description :

Le contenu de l'adresse ou de l'accumulateur spécifié est diminué de 1. Le résultat est remis à la source.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			4A			6A	7A	5A				
OCTETS			1			2	3	1				
CYCLES			2			7	6*	2				
bb			00			10	11	01				

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	*	

* mis à 1 si Q valait 80 hex avant l'opération.

Codes opération :

ACCUM. A: 01001010
bb = 00 HEX = 4A CYCLES = 2

ACCUM. B: 01011010
bb = 01 HEX = 5A CYCLES = 2

INDEXE: 01101010 | DEPL
bb = 10 HEX = 6A CYCLES = 7

ABSOLU: 01111010 | ADRESSE 16 BITS
bb = 11 HEX = 7A CYCLES = 6

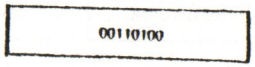
DES

Décrémenter SP

Fonction :

SP ← (SP) − 1

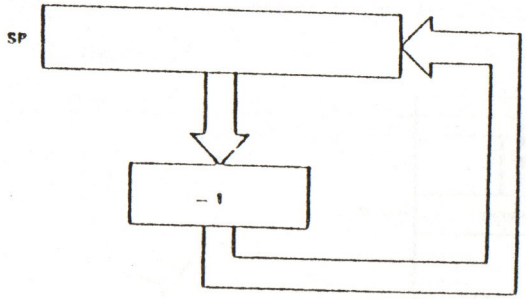
Format :



Description :

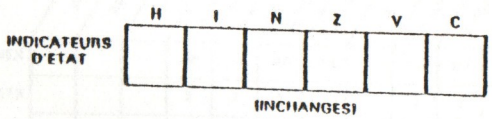
Le contenu de SP est diminué de 1. Permet d'utiliser SP comme compteur ou de se déplacer dans la pile.

Circuit des informations



Mode d'adressage :

Uniquement inhérent
HEX = 34, octet = 1, cycles = 4



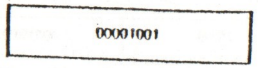
DEX

Décrémenter IX

Fonction :

IX ← (IX) − 1

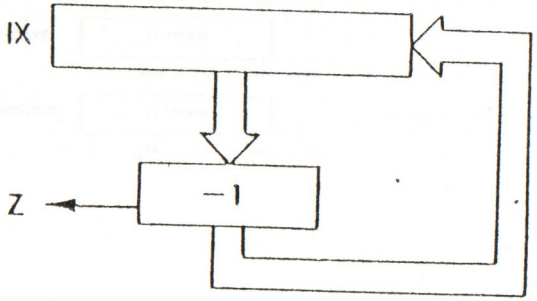
Format :



Description :

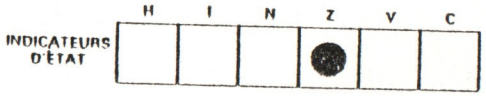
Le contenu de IX est diminué de 1. Permet d'utiliser IX comme compteur. Seul l'indicateur Z est positionné.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent
HEX = 09, octet = 1, cycles =



EOR

Ou exclusif avec l'accumulateur

Fonction :

R ← (R) ⊕ DONNEE

Format :

1abb1000	ADR/DONNEE	ADR

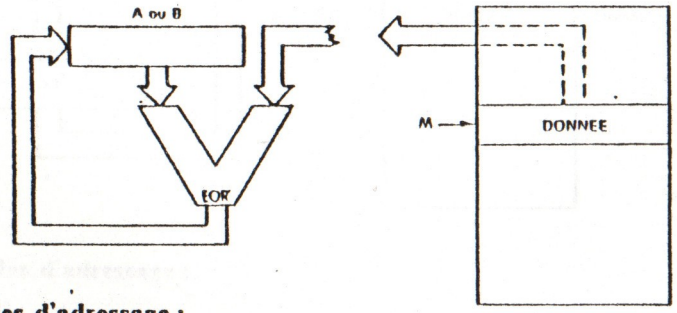
Description :

On effectue le ou exclusif entre la donnée indiquée et l'accumulateur spécifié. La table de vérité est :

	0	1
0	0	1
1	1	0

N.B. : EOR avec « 1 » peut servir à complémenter.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				88	98	A8	B8		C8	D8	E8	F8
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	5	4		2	3	5	4
abb				000	001	010	011		100	101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	○	

Codes opération :

A	IMMEDIAT	10001000	DON	abb = 000 HEX = 88 CYCLES = 2
A	PAGE 0	10011000	ADR	abb = 001 HEX = 98 CYCLES = 3
A	INDEXE	10101000	DEPL	abb = 010 HEX = A8 CYCLES = 5
A	ABSOLU	10111000	ADRESSE 16 BITS	abb = 011 HEX = B8 CYCLES = 4
B	IMMEDIAT	11001000	DON	abb = 100 HEX = C8 CYCLES = 2
B	PAGE 0	11011000	ADR	abb = 101 HEX = D8 CYCLES = 3
B	INDEXE	11101000	DEPL	abb = 110 HEX = E8 CYCLES = 5
B	ABSOLU	11111000	ADRESSE 16 BITS	abb = 111 HEX = F8 CYCLES = 4

INC

Incrémenter en mémoire

Fonction :

Q ← (Q) + 1

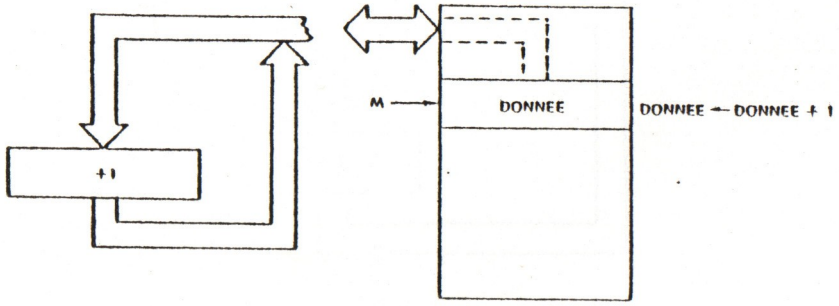
Format :

01bb1100	ADR	ADR

Description :

Le contenu de l'adresse ou de l'accumulateur indiqué est augmenté de 1. Le résultat est remis à la source.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX *	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			4C			6C	7C	5C				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	*	

* mis à 1 si Q valait 7 F avant l'opération.

Codes opération :

ACCUM. A: 01001100
bb = 00 HEX = 4C CYCLES = 2

ACCUM. B: 01011100
bb = 01 HEX = 5C CYCLES = 2

INDEXE: 01101100 | DEPL
bb = 10 HEX = 6C CYCLES = 7

ABSOLU: 01111100 | ADRESSE 16 BITS
abb = 11 HEX = 7C CYCLES = 6

INS — Incrémenter SP

Fonction :

SP ← (SP) + 1

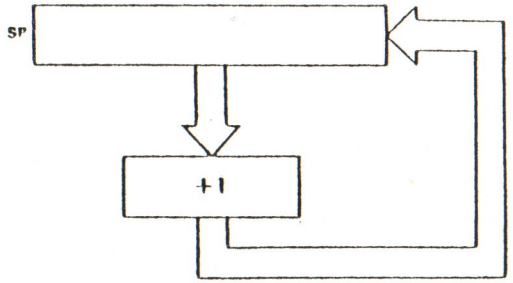
Format :

00110001

Description :

Le contenu de SP est augmenté de 1. Ceci permet d'utiliser SP comme compteur ou de se déplacer dans la pile.

Circuit des informations :



Modes d'adressage :

Uniquement inhérent
HEX = 31, octet = 1, cycles = 4

	H	I	N	Z	V	C
INDICATEURS D'ETAT						

(INCHANGES)

INX — Incrémenter IX

Fonction :

IX ← (IX) + 1

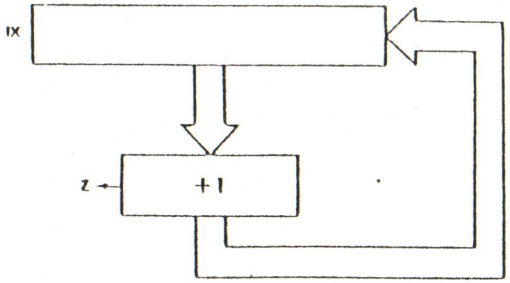
Format :

00001000

Description :

Le contenu de IX est augmenté de 1. Ceci permet d'utiliser IX comme compteur. Le seul indicateur modifié est Z.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent
HEX = 08, octet = 1, cycles = 4

	H	I	N	Z	V	C
INDICATEURS D'ETAT				●		

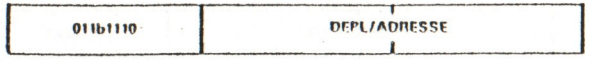
JMP

Saut à une adresse

Fonction :

PC ← ADRESSE

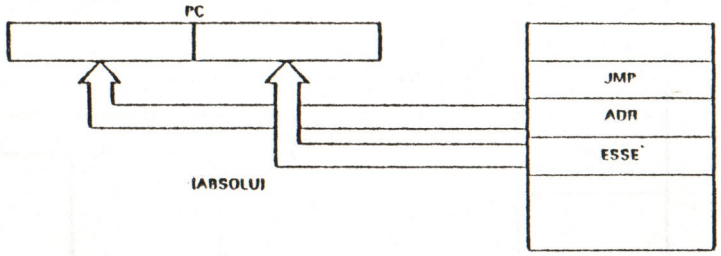
Format :



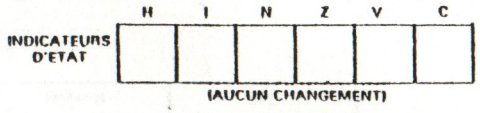
Description :

On charge une nouvelle adresse dans le compteur ordinal, ce qui entraine une rupture de séquence du programme. On peut spécifier une adresse absolue ou indexée.

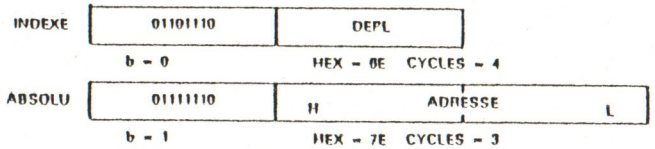
Circuit des informations :

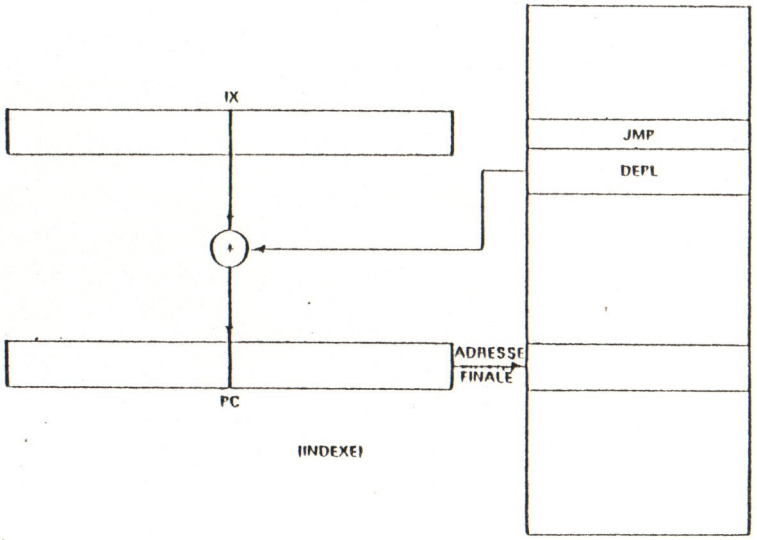


Modes d'adressage : absolu et indexé.



Codes opération :





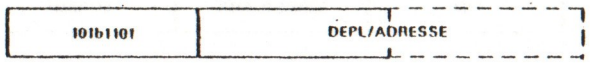
JSR

Saut à un sous-programme

Fonction :

PILE ← (PC) + 2
PC ← ADRESSE

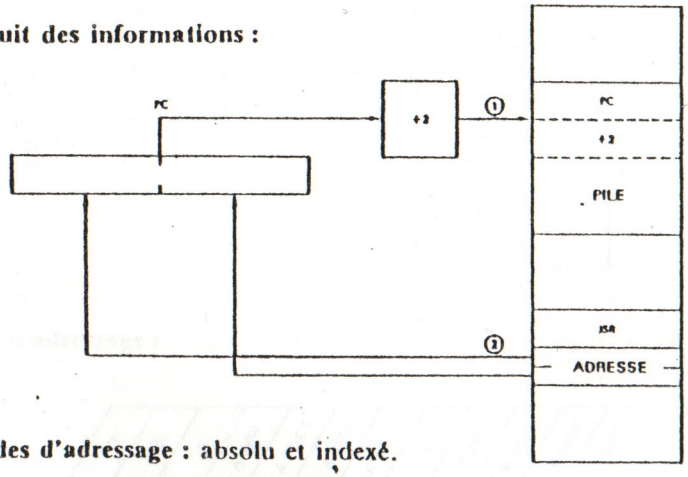
Format :



Description :

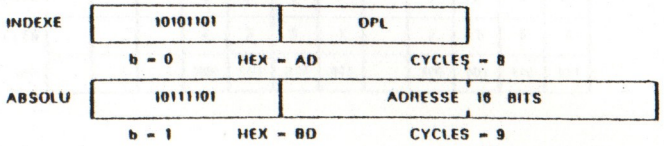
Le contenu du compteur ordinal (+ 2) est sauvé dans la pile. (C'est l'adresse de l'instruction qui suit le JSR). L'adresse du sous-programme est ensuite chargée dans PC. On l'appelle aussi « appel de sous-programme ». Comme pour JMP, l'adresse-cible peut être absolue ou indexée.

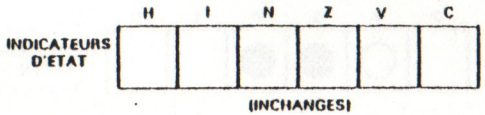
Circuit des informations :



Modes d'adressage : absolu et indexé.

Codes opération :





LDA — Chargement accumulateur

Fonction :

R ← DONNEE

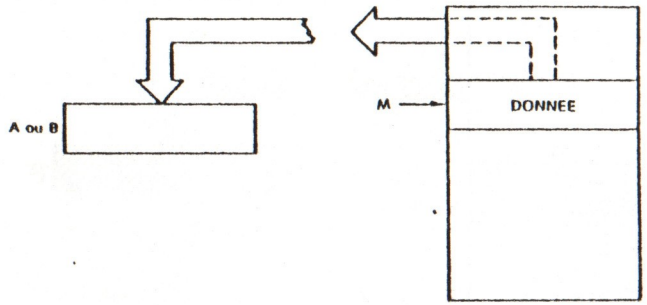
Format :

1abb0110	ADR/DONNEE	ADR

Description :

On charge une nouvelle donnée dans l'accumulateur A ou B.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				B6	96	A6	B6		C6	D6	E6	F6
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	5	4		2	3	5	4
abb				000	001	010	011		100	101	110	111

INDICATEURS D'ETAT	H	I	N	Z	V	C
			●	●	○	

Codes opération :

A	IMMEDIAT	10000110	DON	abb = 000 HEX = 86 CYCLES = 2
A	PAGE 0	10010110	ADR	abb = 001 HEX = 96 CYCLES = 3
A	INDEXE	10100110	DEPL	abb = 010 HEX = A6 CYCLES = 5
A	ABSOLU	10110110	ADRESSE 16 BITS	abb = 011 HEX = B6 CYCLES = 4
B	IMMEDIAT	11000110	DON	abb = 100 HEX = C6 CYCLES = 2
B	PAGE 0	11010110	ADR	abb = 101 HEX = D6 CYCLES = 3
B	INDEXE	11100110	DEPL	abb = 110 HEX = E6 CYCLES = 5
B	ABSOLU	11110110	ADRESSE 16 BITS	abb = 111 HEX = F6 CYCLES = 4

LDS — Chargement du pointeur de pile

Fonction :

SP ← DONNEE (SPH ← (M) ; SPL ← (M'))

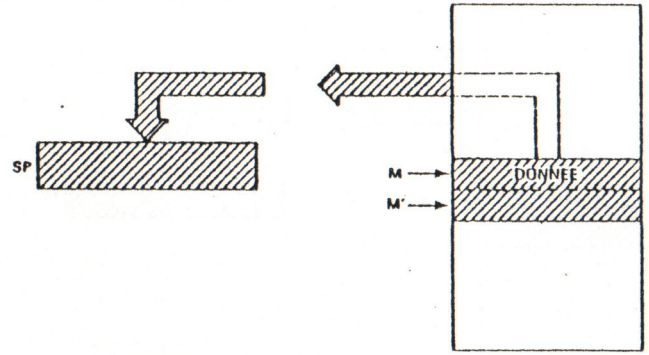
Format :

10bb1110	ADR/DONNEE	ADR/DONNEE

Description :

On charge le registre SP avec la donnée 16 bits qui se trouve à l'adresse indiquée et à la suivante.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				8E	9E	AE	BE					
OCTETS				3	2	2	3					
CYCLES				3	4	6	5					
bb				00	01	10	11					

	H	I	N	Z	V	C
INDICATEURS D'ETAT			*	●	○	

* signe du nombre de 16 bits = bit 15.

Codes opération :

IMMEDIAT	10001110	DONNEE 16 BITS	
	bb = 00	HEX = 8E	CYCLES = 3
PAGE 0	10011110	ADR	
	bb = 01	HEX = 9E	CYCLES = 4
INDEXE	10101110	DEPL	
	bb = 10	HEX = AE	CYCLES = 6
ABSOLU	10111110	ADRESSE 16 BITS	
	bb = 11	HEX = BE	CYCLES = 5

LDX — Chargement du registre index IX

Fonction :

IX ← DONNEE (IXH ← (M) ; IXL ← (M'))

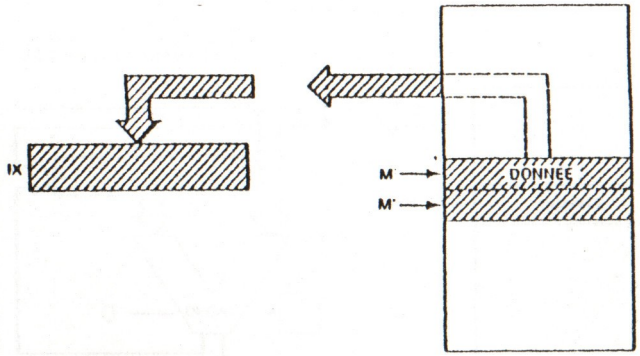
Format :

11bb1110	ADR/DONNÉE	ADR/DONNEE

Description :

On charge le registre IX avec la donnée 16 bits trouvée à l'adresse indiquée et à la suivante.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				CE	DE	EE	FE					
OCTETS				3	2	2	3					
CYCLES				3	4	6	5					
abb				00	01	10	11					

INDICATEURS D'ETAT	H	I	N	Z	V	C
			*	●	○	

* signe du nombre de 16 bits = bit 15.

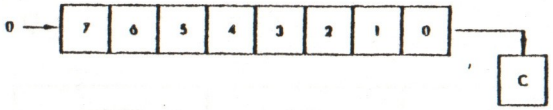
Codes opération :

IMMEDIAT	11001110	DONNNEE	16 BITS
	bb = 00	HEX = CE	CYCLES = 3
PAGE 0	11011110	ADR	
	bb = 01	HEX = DE	CYCLES = 4
INDEXE	11101110	DEPL	
	bb = 10	HEX = EE	CYCLES = 6
ABSOLU	11111110	ADRESSE	16 BITS
	bbb = 11	HEX = FE	CYCLES = 5

LSR

Décalage logique à droite

Fonction :

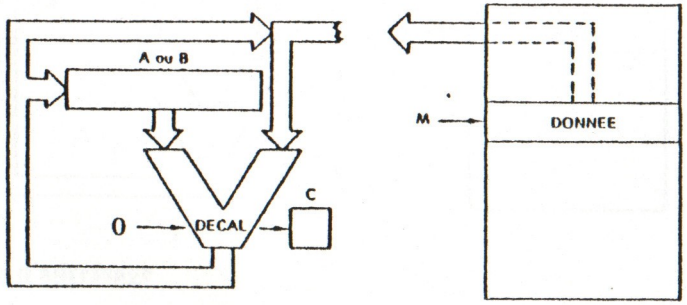


Format :

Description :

On décale d'un cran à droite l'emplacement indiqué (mémoire ou accumulateur). Un 0 est forcé dans le bit 7. Le bit 0 est transféré dans la retenue. La donnée décalée est renvoyée à la source (accumulateur ou mémoire).

Circuit des informations :



Mode d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			44			64	74	64				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

	H	I	N	Z	V	C
INDICATEURS D'ETAT			○	●	*	●

* mis à N ⊕ C (= ici C) après décalage.

Codes opération :

ACCUM. A	01000100		
	bb = 00	HEX = 44	CYCLES = 2
ACCUM. B	01010100		
	bb = 01	HEX = 54	CYCLES = 2
INDEXE	01100100	DEPL	
	bb = 10	HEX = 64	CYCLES = 7
ABSOLU	01110100	ADRESSE 16 BITS	
	bb = 11	HEX = 74	CYCLES = 6

NEG

Complément (à 2)

Fonction :

Q ← 00 — Q

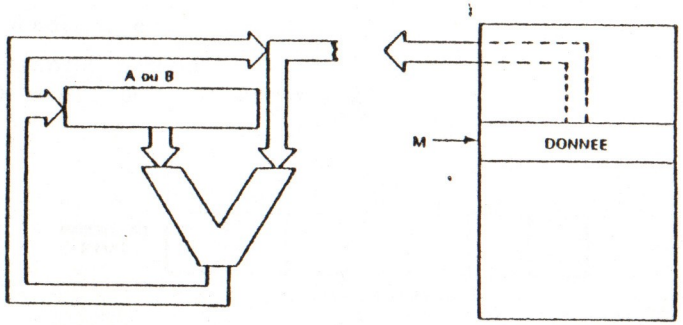
Format :

01bb0000	ADR.	ADR.

Description :

On complémente (à deux) le contenu de l'accumulateur ou de l'emplacement mémoire spécifié, ce qui revient à prendre l'opposé. Le résultat est déposé dans la source : accumulateur ou mémoire.

Circuit des informations



Modes d'adressage

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			40			60	70	50				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	*	*

* V est mis à 1 si Q contenait 80 HEX, sinon à 0.
* C est mis à 0 si Q contenait 00, sinon à 1.

Codes opération :

ACCUM. A: 01000000
bb = 00 HEX = 40 CYCLES = 2

ACCUM. B: 01010000
bb = 01 HEX = 50 CYCLES = 2

INDEXE: 01100000 | DEPL
bb = 10 HEX = 60 CYCLES = 7

ABSOLU: 01110000 | ADRESSE 16 BITS
abb = 11 HEX = 70 CYCLES = 6

NOP

Instruction muette

Fonction :

Aucune

Format :

00000001

Description :

On ne fait rien mais cela dure deux cycles. Peut servir pour rallonger des boucles de délai, ou pour remplir des trous créés par des corrections de programme.

Mode d'adressage :

Uniquement inhérent
HEX = 01, octet = 1, cycles = 2

	N	V	B	D	I	Z	C
INDICATEURS D'ETAT							

(AUCUN CHANGEMENT)

ORA

Ou inclusif avec l'accumulateur

Fonction :

R ← (R) V DONNEE

Format :

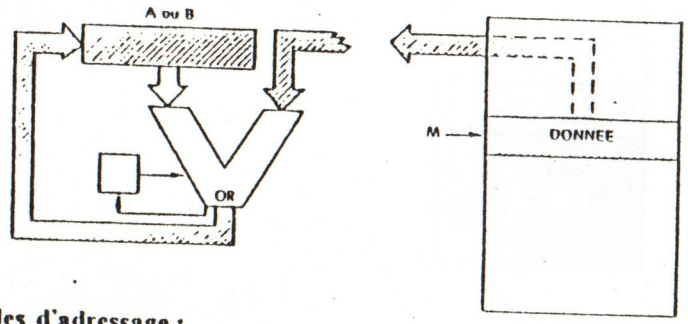
1abb1010	ADR/DONNEE	ADR

Description :

On effectue le OU inclusif de l'accumulateur spécifié et de la donnée indiquée. Le résultat est rangé dans l'accumulateur. Cela peut servir à forcer à 1 des bits choisis.

	0	1
0	0	1
1	1	1

Circuit des informations :



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				BA	9A	AA	BA		CA	DA	EA	FA
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	5	4		2	3	5	4
abb				000	001	010	011		100	101	110	111

INDICATEURS D'ETAT	H	I	N	Z	V	C
			●	●	○	

Codes opération :

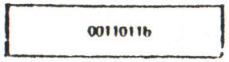
A	IMMEDIAT	10001010	DON	abb = 000 HEX = BA CYCLES = 2
A	PAGE 0	10011010	ADR	abb = 001 HEX = 9A CYCLES = 3
A	INDEXE	10101010	DEPL	abb = 010 HEX = AA CYCLES = 5
A	ABSOLU	10111010	ADRESSE 16 BITS	abb = 011 HEX = BA CYCLES = 4
B	IMMEDIAT	11001010	DON	abb = 100 HEX = CA CYCLES = 2
B	PAGE 0	11011010	ADR	abb = 101 HEX = DA CYCLES = 3
B	INDEXE	11101010	DEPL	abb = 110 HEX = EA CYCLES = 5
B	ABSOLU	11111010	ADRESSE 16 BITS	abb = 111 HEX = FA CYCLES = 4

PSH — Empiler Accumulateur

Fonction :

PILE ← (R)
SP ← (SP) − 1

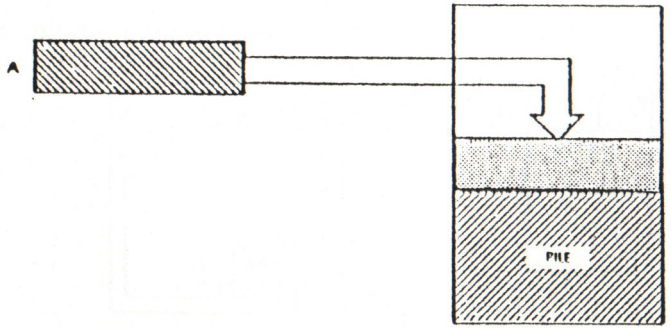
Format :



Description :

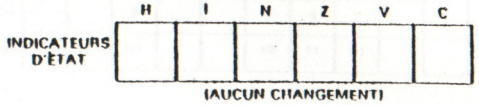
Le contenu de l'accumulateur voulu est mis au sommet de la pile. Le pointeur de pile est remis à jour. L'accumulateur est inchangé.

Circuit des informations :



Modes d'adressage : A ou B.

A : b = 0, HEX = 36, octet = 1, cycles = 4
B : b = 1, HEX = 37, octet = 1, cycles = 4

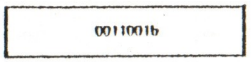


PUL — Dépiler Accumulateur

Fonction :

R ← (PILE)
SP ← (SP) + 1

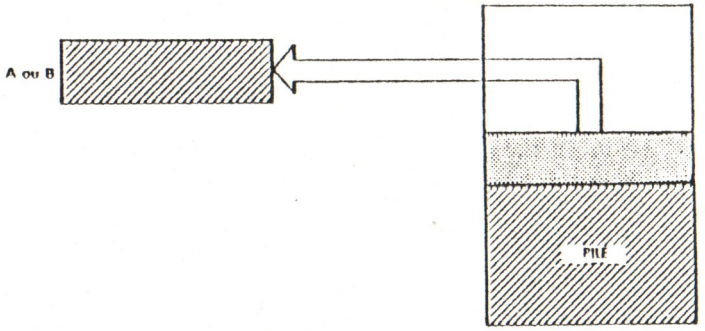
Format :



Description :

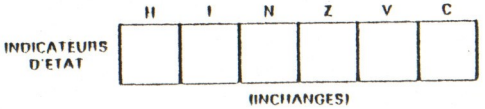
Le sommet de la pile est transféré vers l'accumulateur spécifié. Le pointeur de pile est incrémenté.

Circuit des informations :



Modes d'adressage : A ou B.

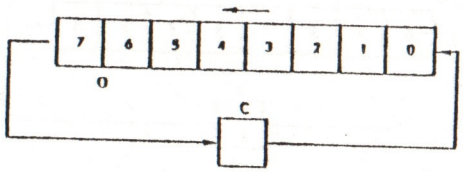
A : b = 0, HEX = 32, octet = 1, cycles = 4
B : b = 1, HEX = 33, octet = 1, cycles = 4



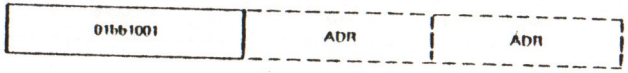
ROL

Rotation à gauche

Fonction :



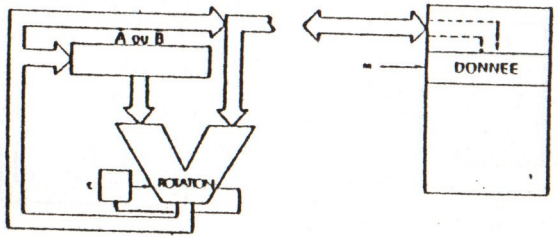
Format :



Description :

Le contenu de l'adresse indiquée (accumulateur ou mémoire) est tourné d'un cran à gauche. La retenue entre dans le bit 0. Le bit 7 devient la nouvelle valeur de la retenue. C'est donc une rotation sur 9 bits.

Circuit des informations :



Mode d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			49			69	79	59				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

INDICATEURS D'ETAT	H	I	N	Z	V	C
			●	●	*	●

* mis à N ⊕ C après rotation.

Codes opération :

ACCUM. A: 01001001
bb = 00 HEX = 49 CYCLES = 2

ACCUM. B: 01011001
bb = 01 HEX = 59 CYCLES = 2

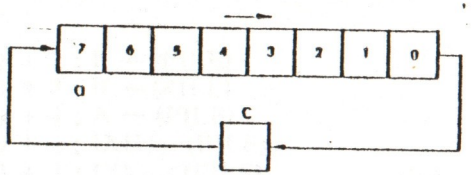
INDEXE: 01101001 | DEPL
bb = 10 HEX = 69 CYCLES = 7

ABSOLU: 01111001 | ADRESSE 16 BITS
abb = 11 HEX = 79 CYCLES = 6

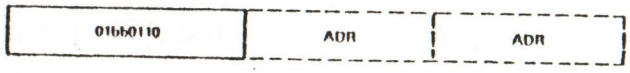
ROR

Rotation à droite

Fonction :



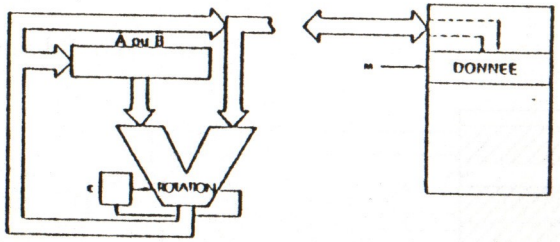
Format :



Description :

Le contenu de l'adresse indiquée (accumulateur ou mémoire) est tourné d'un cran à droite. La retenue va dans le bit 7. Le bit 0 devient la nouvelle valeur de la retenue. C'est donc une rotation sur 9 bits.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			46			66	76	56				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
bb			00			10	11	01				

INDICATEURS D'ETAT	H	I	N	Z	V	C
			●	●	*	●

* mis à N ⊕ C après rotation.

Codes opération :

ACCUM. A	01000110		
	bb = 00	HEX = 46	CYCLES = 2
ACCUM. B	01010110		
	bb = 01	HEX = 56	CYCLES = 2
INDEXE	01100110	DEPL	
	bb = 10	HEX = 66	CYCLES = 7
ABSOLU	01110110	ADRESSE 16 BITS	
	bb = 11	HEX = 76	CYCLES = 6

RTI — Retour d'interruption

Fonction :

SP ← (SP) + 1 ; P ← (PILE)
SP ← (SP) + 1 ; B ← (PILE)
SP ← (SP) + 1 ; A ← (PILE)
SP ← (SP) + 1 ; IXH ← (PILE)
SP ← (SP) + 1 ; IXL ← (PILE)
SP ← (SP) + 1 ; PCH ← (PILE)
SP ← (SP) + 1 ; PCL ← (PILE)

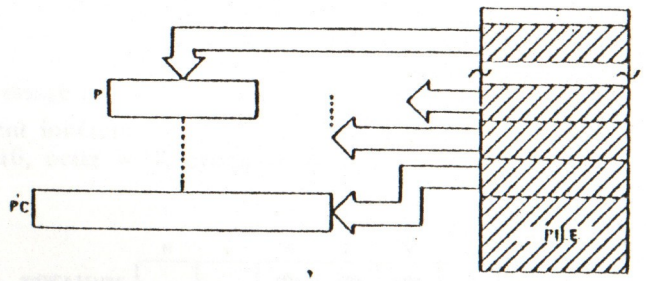
Format :

00111011

Description :

Restaure les registres et le compteur ordinal (PC) qui avaient été sauvés dans la pile. Remet à jour le pointeur de pile.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent
HEX = 3B, octet = 1, cycles = 10

RTS — Retour de sous-programme

Fonction :

PC ← (PILE) (2 octets)
SP ← (SP) + 2

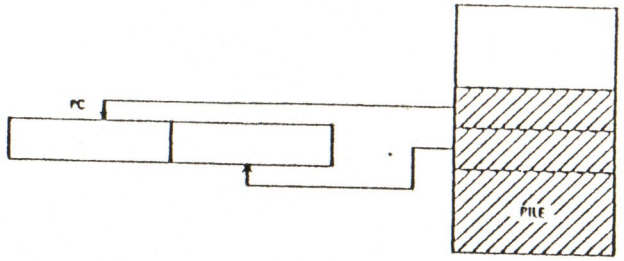
Format :

00111001

Description :

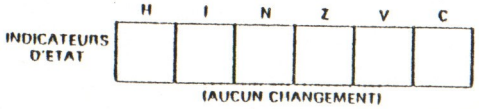
Restaure le compteur ordinal à partir de la pile. Remet à jour le pointeur de pile.

Circuit des informations :



Modes d'adressage :

Uniquement inhérent
HEX = 39, octet = 1, cycles = 5



SBA — Soustraction de B à A

Fonction :

A ← (A) — (B)

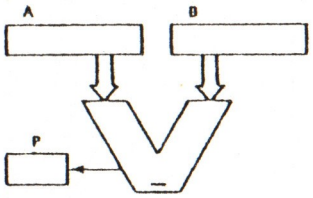
Format :

00010000

Description :

Soustrait de A le contenu de l'accumulateur B. Le résultat est déposé dans l'accumulateur A.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent
HEX = 10, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	●	●

SBC

Soustraction avec retenue

Fonction :

R ← (R) — DONNEE —.C

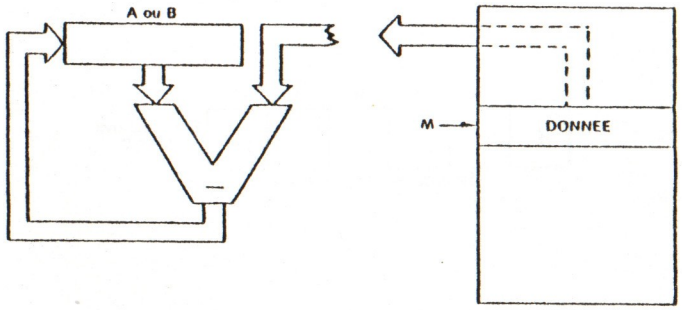
Format :

1abb0010	ADR/DONNEE	ADR

Description :

On soustrait de l'accumulateur spécifié la donnée à l'adresse indiquée, avec la retenue. Le résultat est mis dans l'accumulateur.

Circuit des informations :



Modes d'adressage :

	IMH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				82	92	A2	B2		C2	D2	E2	F2
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	5	4		2	3	5	4
abb				000	001	010	011		100	101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	●	●

Codes opération :

A	IMMEDIAT	10000010	DON	abb = 000 HEX = 82 CYCLES = 2
A	PAGE 0	10010010	ADR	abb = 001 HEX = 92 CYCLES = 3
A	INDEXE	10100010	DEPL	abb = 010 HEX = A2 CYCLES = 5
A	ABSOLU	10110010	ADRESSE 16 BITS	abb = 011 HEX = B2 CYCLES = 4
B	IMMEDIAT	11000010	DON	abb = 100 HEX = C2 CYCLES = 2
B	PAGE 0	11010010	ADR	abb = 101 HEX = D2 CYCLES = 3
B	INDEXE	11100010	DEPL	abb = 110 HEX = E2 CYCLES = 5
B	ABSOLU	11110010	ADRESSE 16 BITS	abb = 111 HEX = F2 CYCLES = 4

SEC

Mise à 1 de la retenue

Fonction :

C ← 1

Format :

00001101

Description :

Le bit de retenue est forcé à 1.

Mode d'adressage :

Uniquement inhérent

HEX = 0D, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT						1

(AUCUN CHANGEMENT)

SEI — Mise à un de l'inhibition interruptions

Fonction :

I ← 1

Format :

00001111

Description :

On met à 1 le masque des interruptions. C'est utilisé lors d'une interruption ou d'une réinitialisation du système.

Mode d'adressage :

Uniquement inhérent

HEX = 0F, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT		1				

SEV — Mise à 1 du débordement

Fonction :

V ← 1

Format :

00001011

Description :

On met à 1 le bit V (débordement) du registre d'état.

Mode d'adressage :

Uniquement inhérent

HEX = 0B, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT					1	

STA Rangement d'un accumulateur en mémoire

Fonction :

M ← (R)

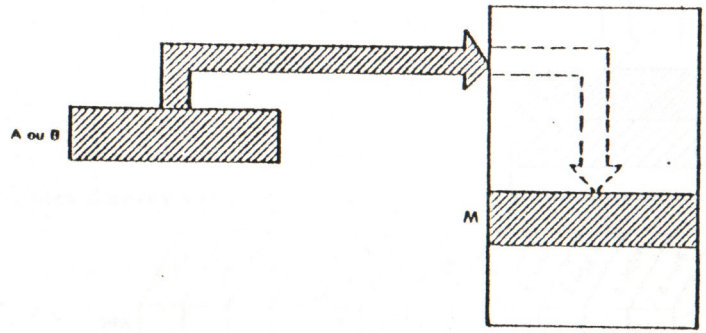
Format :

1abb0111	ADR	ADR

Description :

Le contenu de l'accumulateur voulu est copié dans l'emplacement mémoire indiqué. Le contenu de l'accumulateur concerné est inchangé.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX					97	A7	B7			D7	E7	F7
OCTETS					2	2	3			2	2	3
CYCLES					4	6	5			4	6	5
abb					001	010	011			101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	○	

Codes opération :

A	PAGE 0	10010111	ADR	CYCLES = 4
		abb = 001	HEX = 97	
A	INDEXE	10100111	DEPL	
		abb = 010	HEX = A7	CYCLES = 6
A	ABSOLU	10110111	ADRESSE 16 BITS	
		abb = 011	HEX = B7	CYCLES = 5
B	PAGE 0	11010111	ADR	
		abb = 101	HEX = D7	CYCLES = 4
B	INDEXE	11100111	DEPL	
		abb = 110	HEX = E7	CYCLES = 6
B	ABSOLU	11110111	ADRESSE 16 BITS	
		abb = 111	HEX = F7	CYCLES = 5

STS — Ranger SP en mémoire

Fonction :

(M, M') ← (SP)

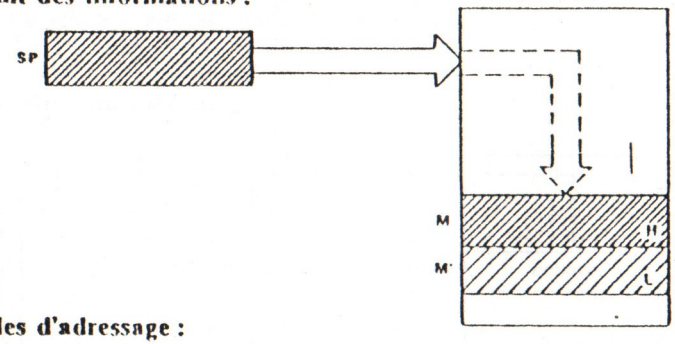
Format :

10bb1111	ADR	ADR

Description :

Le contenu du pointeur de pile est copié dans l'emplacement mémoire indiqué (partie haute) et le suivant (partie basse). Le contenu de SP est inchangé.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX					9F	AF	BF					
OCTETS					2	2	3					
CYCLES					5	7	6					
bb					01	10	11					

INDICATEURS D'ETAT	H	I	N	Z	V	C
			*	●	○	

* signe sur 16 bits = bit 15

Codes opération :

ABSOLU	10111111	ADRESSE	
	bb = 11	HEX = BF	CYCLES = 6
PAGE 0	10011111	ADR	
	bb = 01	HEX = 9F	CYCLES = 5
INDEXE	10101111	DEPL	
	bb = 10	HEX = AF	CYCLES = 7

STX — Ranger IX en mémoire

Fonction :

(M, M') ← IX

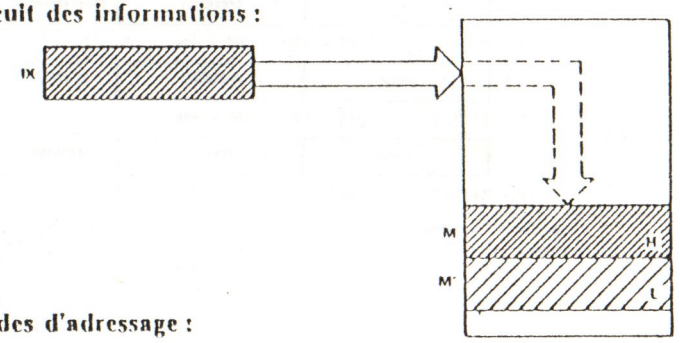
Format :

11bb1111	ADR	ADR

Description :

Le contenu du registre index IX est copié dans l'emplacement mémoire indiqué (partie haute) et le suivant (partie basse). Le contenu de IX est inchangé.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX					DF	EF	FF					
OCTETS					2	2	3					
CYCLES					5	7	6					
bb					01	10	11					

INDICATEURS D'ETAT	H	I	N	Z	V	C
			*	●	○	

* signe sur 16 bits = bit 15.

Codes opération :

ABSOLU	11111111	ADRESSE	
	bb = 11	HEX = FF	CYCLES = 6
PAGE 0	11011111	ADR	
	bb = 01	HEX = DF	CYCLES = 5
INDEXE	11101111	DEPL	
	bb = 10	HEX = EF	CYCLES = 7

SUB

Soustraction

Fonction :

R ← (R) — DONNEE

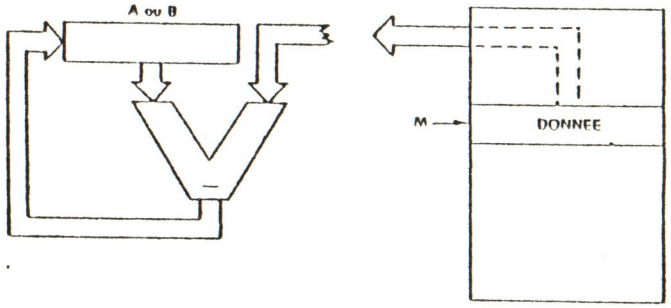
Format :

1abb0000	ADR/DONNEE	ADR

Description :

On soustrait de l'accumulateur spécifié la donnée à l'adresse indiquée. Le résultat est mis dans l'accumulateur.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	A IMM	A PGE 0	A INDX	A ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX				80	90	A0	B0		C0	D0	E0	F0
OCTETS				2	2	2	3		2	2	2	3
CYCLES				2	3	5	4		2	3	5	4
abb				000	001	010	011		100	101	110	111

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	●	●

Codes opération :

A	IMMEDIAT	10000000	DON	
		abb = 000	HEX = 80	CYCLES = 2
A	PAGE 0	10010000	ADR	
		abb = 001	HEX = 90	CYCLES = 3
A	INDEXE	10100000	DEPL	
		abb = 010	HEX = A0	CYCLES = 5
A	ABSOLU	10110000	ADRESSE 16 BITS	
		abb = 011	HEX = B0	CYCLES = 4
B	IMMEDIAT	11000000	DON	
		abb = 100	HEX = C0	CYCLES = 2
B	PAGE 0	11010000	ADR	
		abb = 101	HEX = D0	CYCLES = 3
B	INDEXE	11100000	DEPL	
		abb = 110	HEX = E0	CYCLES = 5
B	ABSOLU	11110000	ADRESSE 16 BITS	
		abb = 111	HEX = F0	CYCLES = 4

SWI — Interruption software

Fonction :

EMPILE (PC) PUIS IX, A, B et P ; PC ← (FFFA, FFFB)

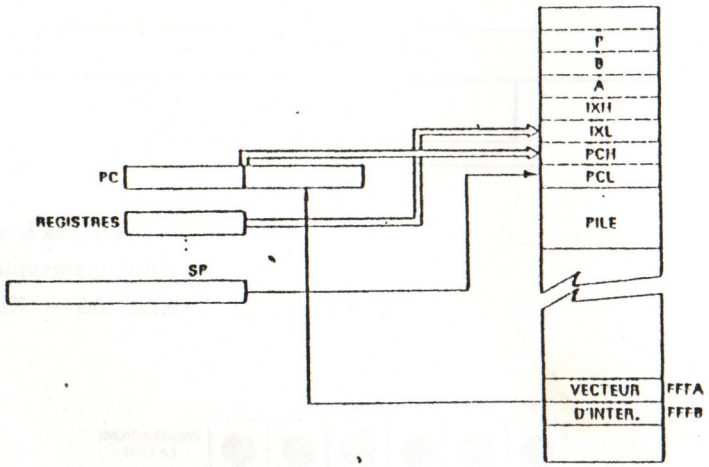
Format :

00111111

Description :

Fonctionne comme une interruption : le compteur ordinal est empilé, ainsi que les registres. Le contenu respectif des cases-mémoire FFFB et FFFA est envoyé dans PCL et PCH.

Circuit des informations



Mode d'adressage :

Uniquement inhérent
HEX = 3F, octet = 1, cycles = 12

	H	I	N	Z	V	C
INDICATEURS D'ETAT		1				

TAB — Transfert de A dans B

Fonction :

B ← (A)

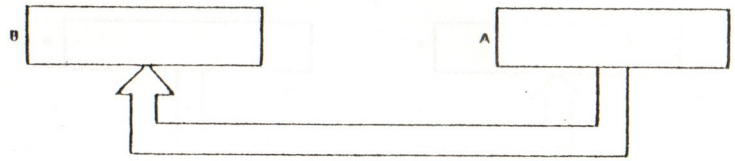
Format :

00010110

Description :

On copie le contenu de l'accumulateur A dans l'accumulateur B. Le contenu de A est inchangé.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent
HEX = 16, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ETAT			●	●	○	

TAP — Transfert de A dans P

Fonction :

P ← (A)

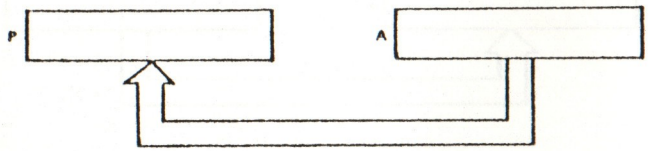
Format :

00000110

Description :

On copie le contenu de l'accumulateur A (bits 0 à 5) dans le registre d'état P. Le contenu de A est inchangé.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent

HEX = 06, octet = 1, cycles = 2

TBA — Transfert de B dans A

Fonction :

A ← (B)

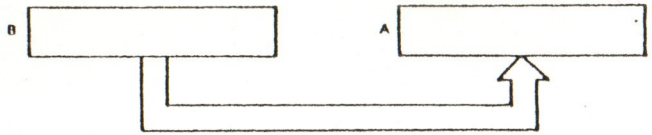
Format :

00010111

Description :

Le contenu de l'accumulateur B est copié dans l'accumulateur A. Le contenu de B est inchangé.

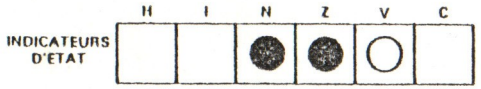
Circuit des informations :



Mode d'adressage :

Uniquement inhérent

HEX = 17, octet = 1, cycles = 2



TPA — Transfert de P dans A

Fonction :

A ← (P)

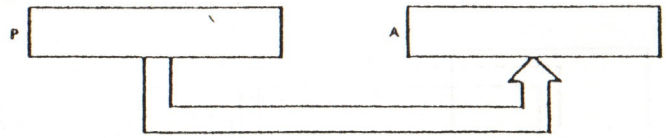
Format :

00000111

Description :

Le contenu du registre d'état P est copié dans l'accumulateur A. Le contenu de P est inchangé. Les bits 6 et 7 de A sont mis à 1.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent

HEX = 07, octet = 1, cycles = 2

	H	I	N	Z	V	C
INDICATEURS D'ÉTAT						

(INCHANGÉS)

TST — Test

Fonction :

(Q) — 00

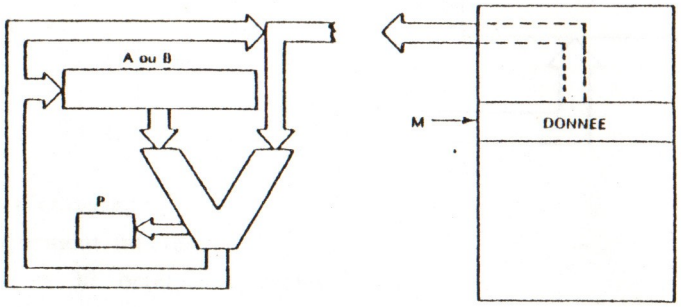
Format :

1bb1101	ADR	ADR

Description :

Positionne les indicateurs N et Z selon la valeur de l'accumulateur ou de la case-mémoire spécifié. N est mis à 1 si Q est négatif et Z est mis à 1 si Q est nul. V et C sont remis à 0.

Circuit des informations :



Modes d'adressage :

	INH	REL	A	IMM	PGE 0	INDX	ABS	B	B IMM	B PGE 0	B INDX	B ABS
HEX			4D			6D	7D	6D				
OCTETS			1			2	3	1				
CYCLES			2			7	6	2				
abb			00			10	11	01				

INDICATEURS D'ETAT	H	I	N	Z	V	C
			●	●	○	○

Codes opérations :

ACCUM. A — 01001101
bb = 00 HEX = 4D CYCLES = 2

ACCUM. B — 01011101
bb = 01 HEX = 5D CYCLES = 2

INDEXE — 01101101 | DEPL
bb = 10 HEX = 6D CYCLES = 7

ABSOLU — 01111101 | ADRESSE 16 BITS
abb = 11 HEX = 7D CYCLES = 6

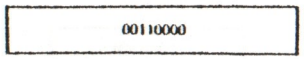
TSX

Transfert de SP dans IX

Fonction :

IX ← (SP) + 1*

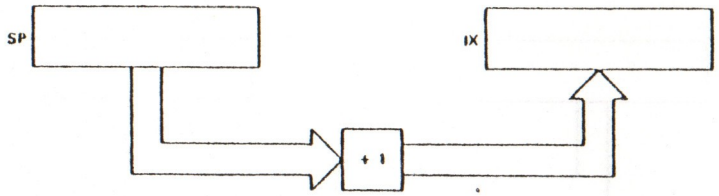
Format :



Description :

Le contenu du pointeur de pile SP augmenté de 1 est copié dans le registre index IX. Le contenu de SP est inchangé.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent

HEX = 30, octet = 1, cycles = 4

	H	I	N	Z	V	C
INDICATEURS D'ETAT						

(INCHANGES)

* ainsi la séquence TSX
LDA A 0, X
est équivalente à PUL A

TXS

Transfert de IX dans SP

Fonction :

SP ← (IX) — 1*

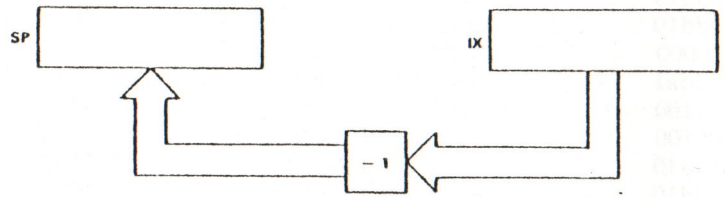
Format :

00110101

Description :

Le contenu du registre index IX diminué de 1 est copié dans le pointeur de pile SP. Le contenu de IX est inchangé.

Circuit des informations :



Mode d'adressage :

Uniquement inhérent

HEX = 35, octet = 1, cycles = 4

	H	I	N	Z	V	C
INDICATEURS D'ETAT						

(AUCUN CHANGEMENT)

* afin que cette instruction soit exactement l'inverse de TSX

WAI

Attente d'interruption

Fonction :

EMPILE PC PUIS IX, A, B et P ;
HALTE LE PROCESSEUR

Format :

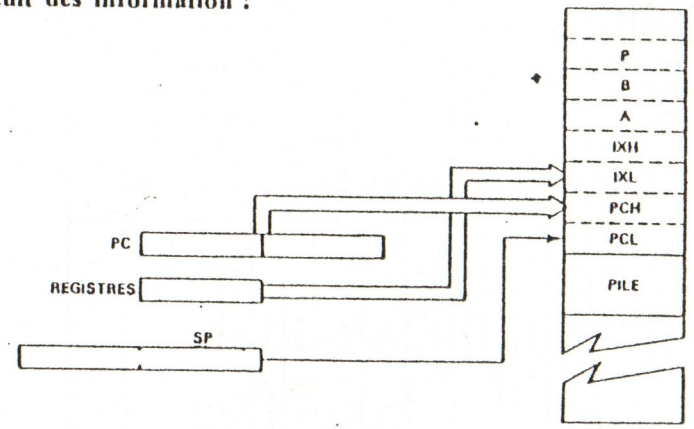
00111110

Description :

Fonctionne comme le début d'une séquence d'interruption : on empile PC et les registres. Puis le processeur se met en attente de la prochaine interruption IRQ. On fera alors PC ← (FFF8, FFF9).

Grâce à WAI, on répond plus vite à une demande d'interruption puisqu'on gagne le temps de rangement des registres, fait d'avance.

Circuit des information :



Mode d'adressage :

Uniquement inhérent

HEX = 3E, octet = 1, cycles = 9

	H	I	N	Z	V	C
INDICATEURS D'ETAT		1*				

* sera mis à 1 lors de l'arrivée de l'interruption.

LISTE DES CODES BINAIRES DES INSTRUCTIONS DU 6800

ABA	00011011	INC	01bb1100
ADC	1abb1001	INS	00110001
ADD	1abb1011	INX	00001000
AND	1abb0100	JMP	011b1110
ASL	01bb1000	JSR	101b1101
ASR	01bb0111	LDA	1abb0110
BCC	00100100	LDS	10bb1110
BCS	00100101	LDX	11bb1110
BEQ	00100111	LSR	01bb0100
BGE	00101100	NEG	01bb0000
BGT	00101110	NOP	00000001
BHI	00100010	ORA	1abb1010
BIT	1abb0101	PSH	0011011b
BLE	00101111	PUL	0011001b
BLS	00100011	ROL	01bb1001
BLT	00101101	ROR	01bb0110
BMI	00101011	RTI	00111011
BNE	00100110	RTS	00111001
BPL	00101010	SBA	00010000
BRA	00100000	SBC	1abb0010
BSR	10001101	SEC	00001101
BVC	00101000	SEI	00001111
BVS	00101001	SEV	00001011
CBA	00010001	STA	1abb0111
CLC	00001100	STS	10bb1111
CLI	00001110	STX	11bb1111
CLR	01bb1111	SUB	1abb0000
CLV	00001010	SWI	00111111
CMP	1abb0001	TAB	00010110
COM	01bb0011	TAP	00000110
CPX	10bb1100	TBA	00010111
DAA	00011001	TPA	00000111
DEC	01bb1010	TST	01bb1101
DES	00110100	TSX	00110000
DEX	00001001	TXS	00110101
EOR	1abb1000	WAI	00111110

LISTE PAR CODE OPERATION HEXADECIMAL (IJ)

I \ J	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		NOP					TAP	TPA	INX	DEX	CLV	SEV	CLC	SEC	CLI	SEI
1	SBA	CBA					TAB	TBA		DAA		ABA				
2	BRA		BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BGE
3	TSX	INS	PUL A	PUL B	DES	TXS	PSH A	PSH B		RTS		RTI			WAI	SWI
4	NEG A			COM A	LSR A		ROR A	ASR A	ASL A	ROL A	DEC A		INC A	TST A		CLR A
5	NEG B			COM B	LSR B		ROR B	ASR B	ASL B	ROL B	DEC B		INC B	TST B		CLR B
6	NEG INDX			COM INDX	LSR INDX		ROR INDX	ASR INDX	ASL INDX	ROL INDX	DEC INDX		INC INDX	TST INDX		CLR INDX
7	NEG ABS			COM ABS	LSR ABS		ROR ABS	ASR ABS	ASL ABS	ROL ABS	DEC ABS		INC ABS	TST ABS		CLR ABS
8	SUB A IMM	CMP A IMM	SBC A IMM		AND A IMM	BIT A IMM	LDA A IMM		EOR A IMM	ADC A IMM	ORA A IMM	ADD A IMM	CPX IMM	BSR	LDS IMM	
9	SUB A PAGE Z	CMP A PAGE Z	SBC A PAGE Z		AND A PAGE Z	BIT A PAGE Z	LDA A PAGE Z	STA A PAGE Z	EOR A PAGE Z	ADC A PAGE Z	ORA A PAGE Z	ADD A PAGE Z	CPX PAGE Z		LDS PAGE Z	STS PAGE Z
A	SUB A INDX	CMP A INDX	SBC A INDX		AND A INDX	BIT A INDX	LDA A INDX	STA A INDX	EOR A INDX	ADC A INDX	ORA A INDX	ADD A INDX	CPX INDX	JSR INDX	LDS INDX	STS INDX
B	SUB A ABS	CMP A ABS	SBC A ABS		AND A ABS	BIT A ABS	LDA A ABS	STA A ABS	EOR A ABS	ADC A ABS	ORA A ABS	ADD A ABS	CPX ABS	JSR ABS	LDS ABS	STS ABS
C	SUB B IMM	CMP B IMM	SBC B IMM		AND B IMM	BIT B IMM	LDA B IMM		EOR B IMM	ADC B IMM	ORA B IMM	ADD B IMM			LDX IMM	
D	SUB B PAGE Z	CMP B PAGE Z	SBC B PAGE Z		AND B PAGE Z	BIT B PAGE Z	LDA B PAGE Z	STA B PAGE Z	EOR B PAGE Z	ADC B PAGE Z	ORA B PAGE Z	ADD B PAGE Z			LDX PAGE Z	STX PAGE Z
E	SUB B INDX	CMP B INDX	SBC B INDX		AND B INDX	BIT B INDX	LDA B INDX	STA B INDX	EOR B INDX	ADC B INDX	ORA B INDX	ADD B INDX			LDX INDX	STX INDX
F	SUB B ABS	CMP B ABS	SBC B ABS		AND B ABS	BIT B ABS	LDA B ABS	STA B ABS	EOR B ABS	ADC B ABS	ORA B ABS	ADD B ABS			LDX ABS	STX ABS

TABLE DE CONVERSION HEXADECIMALE

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	00	000
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	0
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	256	4096
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	512	8192
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	768	12288
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	1024	16384
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	1280	20480
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	1536	24576
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	1792	28672
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	2048	32768
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	2304	36864
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	2560	40960
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	2816	45056
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	3072	49152
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	3328	53248
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	3584	57344
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	3840	61440

5		4		3		2		1		0	
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15

TABLES DE BRANCHEMENTS RELATIFS

BRANCHEMENTS RELATIFS EN AVANT

MSD \ LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

BRANCHEMENTS RELATIFS EN ARRIERE

MSD \ LSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113
9	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97
A	96	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81
B	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65
C	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
E	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
F	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

ACCUMULATOR AND MEMORY OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED ≠	DIRECT OP	DIRECT ~	DIRECT ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INTHER OP	INTHER ~	INTHER ≠	BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	5 H	4 I	3 N	2 Z	1 V	0 C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	•	•	↕	↕	R	•
Clear	CLR							6F	7	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	↕	↕
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	•	•	↕	↕	↕	↕
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	•	•	↕	↕	R	•
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3				M	•	•	↕	↕	↕	↕
	ROLA													49	2	1	A	•	•	↕	↕	↕	↕
	ROLB													59	2	1	B	•	•	↕	↕	↕	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	↕	↕
	RORA													46	2	1	A	•	•	↕	↕	↕	↕
	RORB													56	2	1	B	•	•	↕	↕	↕	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	↕	↕
	ASLA													48	2	1	A	•	•	↕	↕	↕	↕
	ASLB													58	2	1	B	•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	↕	↕
	ASRA													47	2	1	A	•	•	↕	↕	↕	↕
	ASRB													57	2	1	B	•	•	↕	↕	↕	↕
Shift Right, Logic	LSR							64	7	2	74	6	3				M	•	•	R	↕	↕	↕
	LSRA													44	2	1	A	•	•	R	↕	↕	↕
	LSRB													54	2	1	B	•	•	R	↕	↕	↕
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M	•	•	↕	↕	R	•
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subract Acmltrs	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R

Code d'instruction du 6800

INDEX REGISTER AND STACK POINTER OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED ≠	DIRECT OP	DIRECT ~	DIRECT ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INHER OP	INHER ~	INHER ≠	BOOLEAN/ARITHMETIC OPERATION	5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$(X_H/X_L) - (M/M+1)$	•	•	↕	↕	↕	•
Decrement Index Reg	DEX													09	4	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \to X_H$, $(M+1) \to X_L$	•	•	↕	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \to SP_H$, $(M+1) \to SP_L$	•	•	↕	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \to M$, $X_L \to (M+1)$	•	•	↕	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \to M$, $SP_L \to (M+1)$	•	•	↕	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	SP + 1 → X	•	•	•	•	•	•

JUMP AND BRANCH OPERATIONS	MNEMONIC	RELATIVE OP	RELATIVE ~	RELATIVE ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INHER OP	INHER ~	INHER ≠	BRANCH TEST	5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If > Zero	BGE	2C	4	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If < Zero	BLE	2F	4	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2											•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3					•	•	•	•	•	•
No Operation	NOP										01	2	1	Ad Prog Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI										3B	10	1	See special Operation	(10)					
Return From Subroutine	RTS										39	5	1		•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI										3E	9	1		•	(11)	•	•	•	•

CONDITIONS CODE REGISTER OPERATIONS	MNEMONIC	INHER OP	INHER ~	INHER ≠	BOOLEAN OPERATION	5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	A → CCR	(12)					
CCR → Acmlt A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

CONDITION CODE REGISTER NOTES

(Bit set if test is true and cleared otherwise)

(1) (Bit V) Test Result = 10000000 ?
(2) (Bit C) Test Result = 00000000 ?
(3) (Bit C) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)
(4) (Bit V) Test Operand = 10000000 prior to execution ?
(5) (Bit V) Test Operand = 01111111 prior to execution ?
(6) (BIT V) Test Sign bit equal to result of N ⊕ C after shift has executed
(7) (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?
(8) (Bit V) Test 2's complement overflow from subtraction of LS bytes ?
(9) (Bit N) Test Result less than zero ? (Bit 15 = 1)
(10) (All) Load Condition Code Register from Stack (See Special Operations)
(11) (Bit I) Set when Interrupt occurs. If previously set, a Non Maskable interrupt is required to exit the wait state.
(12) (ALL) Set according to the contents of Accumulator A

LEGEND

OP	Operation Code (Hexadecimal)	00	Byte = Zero
~	Number of MPU Cycles	H	Half carry from bit 3
≠	Number of Program Bytes	I	Interrupt mask
+	Arithmetic Plus	N	Negative (sign bit)
−	Arithmatic Minus	Z	Zero (byte)
·	Boolean AND	V	Overflow, 2's complement
M_{SP}	Contents of memory location pointed to be Stack Pointer	C	Carry from bit 7
		R	Reset Always
		S	Set Always
+	Boolean Inclusive OR	↕	Test and set if true cleared otherwise
⊕	Boolean Exclusive OR	•	Not Affected
$\overline{M}$	Complement of M	CCR	Condition Code Register
→	Transfer Into	LS	Least Significant
0	Bit = Zero	MS	Most Significant

ACCUMULATOR AND MEMORY OPERATIONS

OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED ≠	DIRECT OP	DIRECT ~	DIRECT ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INHER OP	INHER ~	INHER ≠	BOOLEAN/ARITHMETIC OPERATION (All register labels refer to contents)	5 H	4 I	3 N	2 Z	1 V	0 C
Add	ADDA	8B	2	2	9B	3	2	AB	5	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	5	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add with Carry	ADCA	89	2	2	99	3	2	A9	5	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	5	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	5	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	5	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Bit Test	BITA	85	2	2	95	3	2	A5	5	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	5	2	F5	4	3				B · M	•	•	↕	↕	R	•
Clear	CLR							6F	7	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	5	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	5	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Acmltrs	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	•	•	↕	↕	①↕	②↕
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	①↕	②↕
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	①↕	②↕
Decimal Adjust, A	DAA													19	2	1	Converts Binary Add. of BCD Characters into BCD Format	•	•	↕	↕	↕	③↕
Decrement	DEC							6A	7	2	7A	6	3				M − 1 → M	•	•	↕	↕	④↕	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	⑤↕	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	④↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	5	2	B8	4	3				A⊕M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	5	2	F8	4	3				B⊕M → B	•	•	↕	↕	R	•
Increment	INC							6C	7	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤↕	•
Load Acmltr	LDAA	86	2	2	96	3	2	A6	5	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	5	2	F6	4	3				M → B	•	•	↕	↕	R	•
Or, Inclusive	ORAA	8A	2	2	9A	3	2	AA	5	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	5	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, M_{SP} → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, M_{SP} → B	•	•	•	•	•	•
Rotate Left	ROL							69	7	2	79	6	3				M	•	•	↕	↕	⑥↕	↕
	ROLA													49	2	1	A	•	•	↕	↕	⑥↕	↕
	ROLB													59	2	1	B	•	•	↕	↕	⑥↕	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	⑥↕	↕
	RORA													46	2	1	A	•	•	↕	↕	⑥↕	↕
	RORB													56	2	1	B	•	•	↕	↕	⑥↕	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	⑥↕	↕
	ASLA													48	2	1	A	•	•	↕	↕	⑥↕	↕
	ASLB													58	2	1	B	•	•	↕	↕	⑥↕	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	⑥↕	↕
	ASRA													47	2	1	A	•	•	↕	↕	⑥↕	↕
	ASRB													57	2	1	B	•	•	↕	↕	⑥↕	↕
Shift Right, Logic	LSR							64	7	2	74	6	3				M	•	•	R	↕	⑥↕	↕
	LSRA													44	2	1	A	•	•	R	↕	⑥↕	↕
	LSRB													54	2	1	B	•	•	R	↕	⑥↕	↕
Store Acmltr	STAA				97	4	2	A7	6	2	B7	5	3				A → M	•	•	↕	↕	R	•
	STAB				D7	4	2	E7	6	2	F7	5	3				B → M	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	5	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	5	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Subract Acmltrs	SBA													10	2	1	A − B → A	•	•	↕	↕	↕	↕
Subtr. with Carry	SBCA	82	2	2	92	3	2	A2	5	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	5	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Acmltrs	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	7	2	7D	6	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B − 00	•	•	↕	↕	R	R

INDEX REGISTER AND STACK POINTER OPERATIONS

OPERATIONS	MNEMONIC	IMMED OP	IMMED ~	IMMED ≠	DIRECT OP	DIRECT ~	DIRECT ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INHER OP	INHER ~	INHER ≠	BOOLEAN/ARITHMETIC OPERATION	5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3				$(X_H/X_L) - (M/M + 1)$	•	•	↕	↕	↕	•
Decrement Index Reg	DEX													09	4	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	4	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	4	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	4	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				$M \rightarrow X_H$, $(M + 1) \rightarrow X_L$	•	•	↕	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				$M \rightarrow SP_H$, $(M + 1) \rightarrow SP_L$	•	•	↕	↕	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				$X_H \rightarrow M$, $X_L \rightarrow (M + 1)$	•	•	↕	↕	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				$SP_H \rightarrow M$, $SP_L \rightarrow (M + 1)$	•	•	↕	↕	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	SP + 1 → X	•	•	•	•	•	•

JUMP AND BRANCH OPERATIONS

OPERATIONS	MNEMONIC	RELATIVE OP	RELATIVE ~	RELATIVE ≠	INDEX OP	INDEX ~	INDEX ≠	EXTND OP	EXTND ~	EXTND ≠	INHER OP	INHER ~	INHER ≠	BRANCH TEST	5 H	4 I	3 N	2 Z	1 V	0 C
Branch Always	BRA	20	4	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	4	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	4	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	4	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	4	2										N⊕V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										Z + (N⊕V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	4	2										Z + (N⊕V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										N⊕V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	4	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	4	2										Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	4	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	4	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	4	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	8	2											•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3					•	•	•	•	•	•
No Operation	NOP										01	2	1	Ad Prog Cntr. Only	•	•	•	•	•	•
Return From interrupt	RTI										3B	10	1		——⑩——					
Return From Subroutine	RTS										39	5	1	See special Operation	•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI										3E	9	1		•	⑪	•	•	•	•

CONDITIONS CODE REGISTER OPERATIONS

OPERATIONS	MNEMONIC	INHER OP	INHER ~	INHER ≠	BOOLEAN OPERATION	5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	A → CCR	——⑫——					
CCR → Acmlt A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

CONDITION CODE REGISTER NOTES

(Bit set if test is true and cleared otherwise)

① (Bit V) Test Result = 10000000 ?

② (Bit C) Test Result = 00000000 ?

③ (Bit C) Test Decimal value of most significant BCD Character greater than nine ? (Not cleared if previously set)

④ (Bit V) Test Operand = 10000000 prior to execution ?

⑤ (Bit V) Test Operand = 01111111 prior to execution ?

⑥ (BIT V) Test Set equal to result of N⊕C after shift has occurred

⑦ (Bit N) Test Sign bit of most significant (MS) byte of result = 1 ?

⑧ (Bit V) Test 2's complement overflow from subtraction of LS bytes ?

⑨ (Bit N) Test Result less than zero ? (Bit 15 = 1)

⑩ (All) Load Condition Code Register from Stack (See Special Operations)

⑪ (Bit I) Set when interrupt occurs. If previously set, a Non Maskable interrupt is required to exit the wait state.

⑫ (ALL) Set according to the contents of Accumulator A

LEGEND

OP	Operation Code (Hexadecimal)
~	Number of MPU Cycles
≠	Number of Program Bytes
+	Arithmetic Plus
−	Arithmetic Minus
·	Boolean AND
M_{SP}	Contents of memory location pointed to be Stack Pointer
+	Boolean Inclusive OR
⊕	Boolean Exclusive OR
$\overline{M}$	Complement of M
→	Transfer Into
0	Bit = Zero
00	Byte = Zero
H	Half carry from bit 3
I	Interrupt mask
N	Negative (sign bit)
Z	Zero (byte)
V	Overflow, 2's complement
C	Carry from bit 7
R	Reset Always
S	Set Always
↕	Test and set if true, cleared otherwise
•	Not Affected
CCR	Condition Code Register
LS	Least Significant
MS	Most Significant